

SISTEM PENYEMBUNYIAN FILE DOCUMENT DENGAN MENGUNAKAN *METODE LEAST SIGNIFICANT BIT* (LSB) PADA CITRA BITMAP

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

Oleh :

TRI HANDAYANINGTYAS
10651004396



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2011**

SISTEM PENYEMBUNYIAN FILE DOCUMENT DENGAN MENGUNAKAN *METODE LEAST SIGNIFICANT BIT* (LSB) PADA CITRA BITMAP

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

oleh :

Tri Handayaniingtyas
10651004396



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2011**

KATA PENGANTAR

Assalamu'alaikum, wr,wb,

Syukur Alhamdulillah penulis haturkan kehadiran Allah SWT atas rahmat, nikmat, karunia serta hidayah yang telah dilimpahkan-Nya kepada penulis sehingga dapat menyelesaikan Tugas Akhir dengan judul **“SISTEM PENYEMBUNYIAN FILE DOCUMENT DENGAN MENGGUNAKAN METODE LEAST SIGNIFICANT BIT PADA CITRA BITMAP”** sebagai syarat kelulusan dalam menyelesaikan studi di Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau.

Dalam menyelesaikan Tugas Akhir ini penulis banyak mendapat bimbingan, bantuan baik secara moril maupun materil dan dukungan yang sangat berarti dari berbagai pihak. Untuk itu penulis banyak mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. H. M Nazir selaku Rektor UIN SUSKA yang juga dalam hal ini banyak memberikan bantuan baik secara langsung maupun tidak langsung dalam proses penyelesaian tugas akhir ini.
2. Ibu Dra. Yenita Morena, M.Si selaku Dekan Fakultas Sains & Teknologi yang juga dalam hal ini banyak memberikan bantuan dan dukungan dalam proses penyelesaian tugas akhir ini.
3. Bapak Novriyanto, M.Sc selaku Ketua Jurusan Teknik Informatika yang selalu memberikan bantuan, bimbingan dan dukungan dalam penyelesaian proses tugas akhir ini.
4. Bapak Benny Sukma Negara, ST, MT selaku pembimbing yang selalu memberikan bimbingan dan petunjuk sehingga tugas akhir ini dapat diselesaikan dengan baik.

5. Bapak Surya Agustian, ST, M.Kom selaku Penguji I dan Bapak Febi Yanto M.Kom selaku Penguji II yang telah memberikan masukan yang bermanfaat kepada penulis.
6. (Alm) Ayahanda Willi Parno dan Ibunda Marleyani Yetty, terima kasih banyak atas pengorbanannya memberikan dukungan dan semangat demi kesuksesan dan kejayaan anak-anaknya. Dan selalu menjadi inspirasi, motivasi hidupku dalam setiap langkahku untuk menyelesaikan tugas akhir ini. Semoga beliau dalam lindungan Allah SWT dimana pun berada, dan penulis memohon do'a semoga pengorbanan beliau mendapat keridhoan dari Allah SWT. Amiin...
7. Kedua masku Agus Suwito Haryoko, ST dan Setiono Dwi Haryadi, ST yang selalu membimbing dan mendidik adik-adiknya untuk menjadi lebih baik. Adikku Indah Wulandari serta Nenekku tercinta Suryani, terima kasih atas kasih sayang dan pengorbanan yang selalu diberikan kepada penulis guna menyelesaikan tugas akhir ini.
8. Segenap dosen Teknik Informatika yang tidak dapat saya sebutkan satu persatu yang telah banyak memberikan ilmu dan bimbingan akademis kepada penulis selama masa perkuliahan.
9. Teman-temanku Aank, Ade, Bayu, Jason, Bobby, Rendra, Rafika, Eka, Merry, Mely, Reni, Pipit terima kasih atas motivasi dan kebersamaan yang telah diberikan selama ini.
10. Sahabatku, Fitri, Angga Novanda Putra, G.Suroto dan Jamyla Ilyas yang telah meluangkan waktu untuk membantu dan mendukung Penulis seperti saran, kritik dan diskusi selama melaksanakan dan penyusunan laporan tugas akhir ini.
11. Teman-teman seperjuangan TIF C angkatan 2006, khususnya Rizky, Gatot, Bahrur Roji, Rinto, Zulkifli, terima kasih atas bantuan dan motivasi yang diberikan selama ini. Semoga kalian bisa tamat secepatnya, Amin ya rabbal alamin.
12. Semua teman-teman satu angkatan, satu jurusan dan satu fakultas yang tidak dapat disebutkan satu persatu.

13. Serta pihak-pihak lainnya yang telah turut membantu dalam proses pembuatan tugas akhir ini baik secara langsung maupun tidak langsung.

Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna. Oleh karena itu, kritik serta saran yang membangun dari rekan-rekan pembaca sangat dibutuhkan agar dapat membuat tugas akhir ini lebih baik. Akhir kata penulis berharap agar tugas akhir ini bisa memberikan manfaat bagi pembaca dan semua pihak yang berkepentingan. Terima kasih.

Pekanbaru, Juli 2011

Penulis

SISTEM PENYEMBUNYIAN FILE DOCUMENT DENGAN MENGUNAKAN *METODE LEAST SIGNIFICANT BIT (LSB)* PADA CITRA BITMAP

TRI HANDAYANINGTYAS

10651004396

Tanggal Sidang : 4 Juli 2011
Periode Wisuda : November 2011

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Pengiriman gambar melalui jaringan internet saat ini telah banyak digunakan, namun pengiriman tersebut belum tentu aman. Salah satu cara pengamanan dalam pengiriman gambar adalah dengan teknik steganografi. Steganografi dapat menyembunyikan data dan informasi rahasia kedalam data lain yang tampak tidak mengandung apa-apa.

Aplikasi StegLSB ini dibangun menggunakan metode *least significant bit insertion*. Pengembangan aplikasi steganografi pada citra digital dengan menggunakan metode *least significant bit insertion* ini bekerja dengan menggantikan bit-bit terakhir pada gambar dengan bit data yang berupa file.

Dari penelitian ini diperoleh hasil bahwa mutu gambar yang telah disteganografi tidak mengalami perubahan berarti dan data yang berada dalam gambar dapat diekstrak kembali. Namun gambar tidak tahan terhadap proses manipulasi (pemotongan *frame*) yang dilakukan pada gambar tersebut.

Kata Kunci : Citra Bitmap, *Least Significant Bit*, Steganografi.

HIDING SYSTEM OF DOCUMENT FILES USING THE LEAST SIGNIFICANT BIT (LSB) IN BITMAP IMAGE

TRI HANDAYANINGTYAS

10651004396

Date of Final Exam : July 4th 2011

Graduation Period : November 2011

Informatics Engineering Department

Science and Technology Faculty

State Islamic University of Sultan Syarif Kasim Riau

ABSTRACT

The image transmission through internet is now widely used, but the transmission is not necessarily safe. One way of securing the transmitted images is a steganography techniques. Steganography hides the data and confidential information into another data which seems contain nothing.

StegLSB application was built using the least significant bit insertion. The development of steganography application on a digital image using this least significant bit insertion method works by replacing the least bits in the image with the data bit in the form of file.

This study produce the outcome that the quality of image which has already steganographed is unchanged and the data which is in the image can be re-extracted. But the image is not resistant to the manipulation (frame cutting) which is done to that image.

Keywords : Image Bitmap, Least Significant Bit, Steganography.

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL	iv
LEMBAR PERNYATAAN.....	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK.....	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xvi
DAFTAR TABEL.....	xviii
DAFTAR SIMBOL	xix
DAFTAR RUMUS	xix
 BAB I PENDAHULUAN	 I-1
1.1 Latar Belakang.....	I-1
1.2 Rumusan Masalah.....	I-2
1.3 Batasan Masalah	I-3
1.4 Tujuan.....	I-3
1.5 Sistematika Penulisan	I-3
 BAB II LANDASAN TEORI	 II-1
2.1 Keamanan Data.....	II-1
2.2 Steganografi.....	II-1
2.2.1 Metode Steganografi	II-4
2.3 Least Significan Bit Insertion (LSB)	II-4
2.4 Citra Digital	II-6
2.4.1 Format Citra Digital	II-6

	2.4.2 Citra Bitmap	II-9
	2.5 Steganografi Pada Media Digital	II-11
BAB III	METODOLOGI PENELITIAN	III-1
	3.1 Perumusan Masalah	III-2
	3.2 Studi Pustaka	III-2
	3.3 Analisa	III-2
	3.4 Perancangan Aplikasi	III-3
	3.5 Implementasi	III-4
	3.6 Pengujian	III-4
	3.7 Kesimpulan dan Saran	III-4
BAB IV	ANALISA DAN PERANCANGAN	IV-1
	4.1 Analisis	IV-1
	4.1.1 Analisis Umum.....	IV-1
	4.1.2 Analisis Rinci	IV-2
	4.1.3 Analisa Kebutuhan Sistem	IV-4
	4.1.3.1 Analisis Data.....	IV-4
	4.1.3.2 Analisis Masukan.....	IV-4
	4.1.3.3 Analisa Proses	IV-4
	4.1.3.4 Analisis Keluaran Gambar Yang Disisipkan File.....	IV-4
	4.1.4 Analisa Metode.....	IV-9
	4.2 Perancangan.....	IV-10
	4.2.1 Perancangan Antar Muka Aplikasi	IV-10
	4.2.1.1 Perancangan Antarmuka Proses Penyisipan File	IV-11
	4.2.1.2 Perancangan Antarmuka Proses Ekstraksi.....	IV-15
BAB V	IMPLEMENTASI DAN PENGUJIAN	V-1
	5.1 Implementasi Sistem.....	V-1
	5.1.1 Alasan Pemilihan Perangkat Lunak	V-1



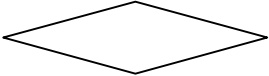
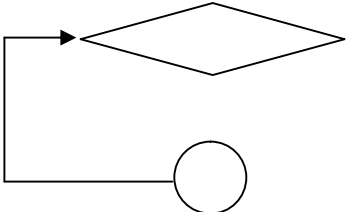


5.1.2	Alasan Pemilihan File BMP	V-1
5.1.3	Batasan Implementasi	V-2
5.1.4	Lingkungan Implementasi	V-2
5.1.5	Tampilan aplikasi.....	V-3
5.2	Pengujian Sistem	V-6
5.2.1	Pengujian Menggunakan <i>Black Box</i>	V-7
5.2.2	Pengujian Modul	V-7
5.2.2.1	Pengujian Modul Penyisipan File.....	V-7
5.2.2.2	Pengujian Modul Pengambilan File	V-8
5.2.3	Pengujian Berdasarkan <i>Fidelity</i>	V-9
5.2.4	Pengujian Berdasarkan <i>Recovery</i>	V-9
5.2.5	Pengujian Berdasarkan Waktu	V-10
5.2.6	Pengujian Berdasarkan PSNR.....	V-18
5.2.3	Pengujian Berdasarkan <i>Robustness</i>	V-19
5.3	Kesimpulan Pengujian.....	V-20
BAB VI	PENUTUP	VI-1
6.1	Kesimpulan.....	VI-1
6.2	Saran.....	VI-2

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR RIWAYAT HIDUP

DAFTAR SIMBOL

Simbol	Keterangan
	Untuk menyatakan START, STOP, RETURN
	Untuk menyatakan proses –proses perhitungan-perhitungan sederhana
	Untuk menyatakan proses membanding, testing pengambilan keputusan
	Untuk menyatakan proses berulang (<i>looping</i>)
	Untuk menyatakan aliran proses
	Untuk menyatakan <i>input/output</i>

DAFTAR TABEL

TABEL	Halaman
5.1 Butir Uji Pengujian Penyisipan File	V-7
5.2 Butir Uji Pengujian Pengambilan File	V-9
5.3 Pengujian Besar File Gambar Hasil Steganografi.....	V-10
5.4 Pengujian Recovery	V-11
5.5 Pengujian Waktu Penyisipan di Bit ke-8.....	V-12
5.6 Pengujian Waktu Penyisipan di Bit ke-7	V-14
5.7 Pengujian Waktu Penyisipan di Bit ke-6.....	V-16
5.8 Pengujian Citra dalam Nilai PSNR.....	V-19
5.9 Pengujian Keberadaan File Dalam Gambar	V-20

DAFTAR RUMUS

TABEL	Halaman
4.1 Rumus MSE	IV-10
4.2 Rumus PSNR	IV-10

DAFTAR GAMBAR

GAMBAR	Halaman
2.1 <i>Steganographic System</i>	II-2
2.2 <i>Graphical Version of a Steganographic System</i>	II-3
2.3 Struktur <i>File</i> BMP	II-9
3.1 Diagram Alir Metodologi Penelitian.....	III-1
4.1 <i>Flowchart</i> Ekstraksi Gambar Ke Bentuk Biner	IV-6
4.2 <i>Flowchart</i> Ekstraksi <i>File</i> Ke Bentuk Biner.....	IV-7
4.3 <i>Flowchart</i> Proses Penyisipan File Kedalam Gambar.....	IV-8
4.4 <i>Flowchart</i> Proses Ekstraksi Hasil	IV-9
4.5 Rancangan Menu Utama	IV-12
4.6 Rancangan Menu Stegano LSB	IV-13
4.7 Rancangan Menu Penyisipan File	IV-14
4.8 Rancangan Kata Kunci Penyisipan	IV-15
4.9 Rancangan Pengambilan File	IV-16
4.10 Rancangan Kata Kunci Pengambilan File.....	IV-17
5.1 Tampilan Menu Utama Aplikasi Steganografi <i>File</i> Pada Gambar	V-3
5.2 Tampilan Menu Pemilihan Bit Penyisipan	V-4
5.3 Tampilan Menu Penyisipan.....	V-5
5.4 Tampilan Menu Pengambilan File	V-6
5.5 Picture1.bmp, Picture2.bmp, Picture3.bmp	V-10
5.6 (1)Picture1.bmp, Gambar(2)Music1.bmp	V-11
5.7 (3)Sunset1.bmp	V-12
5.8 (1)Gambar1.bmp, (2)IMG_1.bmp.....	V-13
5.9 (1)Picture1.bmp, (2) Pacujalur1.bmp.....	V-15
5.10 (1) IMG_9178.bmp, (2) Wedding1.bmp.....	V-17

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi telekomunikasi dan penyimpanan data dengan menggunakan komputer memungkinkan pengiriman data jarak jauh yang relatif cepat dan murah. Keterbukaan informasi di internet sangat memungkinkan pihak lain dapat mengakses data yang dikirim. Untuk itu, diperlukan sistem pengamanan data yang cukup tinggi dalam proses pertukaran informasi tersebut. Penyembunyian pesan rahasia dalam citra digital merupakan salah satu sistem pengamanan data.

Saat ini perkembangan teknik penyembunyian pesan pada citra digital sangat berkembang pesat. Sebelumnya telah ada cara untuk menjaga keamanan data yang dikenal dengan kriptografi. Dengan adanya kriptografi, kerahasiaan data terjaga keamanannya, namun bentuk ciphertext yang acak akan menimbulkan kecurigaan pihak ketiga akan kerahasiaan *file*. Untuk itu, diterapkan steganografi yang dalam bahasa Yunani berarti pesan tersembunyi (*covered writing*) dalam usaha menjaga kerahasiaan data. Steganografi mempunyai keunggulan dengan menyembunyikan keberadaan pesan tersebut. Steganografi menyembunyikan pesan pada media digital sehingga pesan tidak terlihat.

Pada penelitian sebelumnya, aplikasi steganografi teks dengan menggunakan metode *least significant bit* pada citra bitmap telah berhasil dilakukan. Data bitmap yang disisipi data teks dan karakter, tidak mengalami perubahan warna yang cukup berarti sehingga tidak disadari oleh mata manusia bahwa data bitmap tersebut telah disisipi data lain. Kelemahan dari penelitian yang sebelumnya, aplikasi steganografi ini hanya dapat menyisipkan data teks dan karakter (Rehazain, 2007).

Untuk menyembunyikan data pada citra bitmap dengan steganografi digunakan metode *Least Significant Bit* (LSB). Metode LSB memanfaatkan kelemahan mata manusia yang tidak dapat membedakan perubahan yang kecil, karena perubahan terjadi pada bit terakhir sehingga perubahan warna yang dihasilkan tidak penting. Sedangkan media yang digunakan sebagai *file* penampung berupa citra digital untuk mengurangi kecurigaan pihak lain akan keberadaan *file* rahasia yang ingin disembunyikan. Tingkat ketakterlihatan (*invisibility*) pada metode LSB cukup tinggi. Dari penjelasan metode LSB sebelumnya, penulis akan melakukan perubahan pada 3 bit terakhir. Perubahan ini dilakukan untuk membuktikan pengaruh apa saja yang akan terjadi pada gambar yang disisipkan *file* pada 3 bit terakhir yaitu di bit ke 8, bit ke 7, dan bit ke 6.

Pengembangan aplikasi steganografi metode *least significant bit* diperlukan agar *user* dapat menyembunyikan data dari semua jenis file dalam suatu citra bitmap. Selain itu, penulis juga mengembangkan metode *least significant bit* dimana proses penyisipan akan dilakukan di 3 bit terendah pada citra bitmap. Hal ini dilakukan untuk mengetahui perubahan yang terjadi pada citra bitmap yang telah melalui proses steganografi di 3 bit terendah

Dalam tugas akhir ini penulis akan membahas lebih lanjut tentang implementasi pengujian kualitas citra bitmap yang disisipkan *file* rahasia pada 3 bit terakhir dengan pengembangan metode *Least Significant Bit Insertion* (LSB).

1.2 Rumusan Masalah

Sebagaimana telah dipaparkan sebelumnya pada latar belakang, maka didapatkan rumusan masalah dari tugas akhir ini, yaitu: Bagaimana merancang aplikasi penyembunyian file dengan menggunakan metode *Least Significant Bit Insertion* (LSB) di bit 3 terakhir.

1.3 Batasan Masalah

Adapun batasan masalah dalam tugas akhir ini adalah sebagai berikut :

1. Pengembangan metode steganografi yang digunakan adalah metode *Least Significant Bit Insertion* (LSB) yang diterapkan pada 3 bit terakhir.

2. Menggunakan *file* BMP sebagai *carier image* karena file BMP memiliki ukuran yang jauh lebih besar daripada tipe-tipe yang lain.

1.4 Tujuan

Tujuan yang ingin dicapai dalam penyusunan tugas akhir ini adalah:

Mengembangkan pengujian sistem penyembunyian *file document* dengan menggunakan metode Least Significant Bit (LSB) pada citra bitmap.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut:

1. Bab 1 Pendahuluan, berisi berisi latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika penulisan
2. Bab 2 Landasan Teori, menjelaskan tentang landasan teori berdasarkan literatur yang menjadi acuan dalam pelaksanaan skripsi , yaitu steganografi, metode *Least Significant Bit Insertion* (LSB) dan citra digital.
3. Bab 3 Metodologi Penelitian, menjelaskan tahapan dan langkah-langkah penelitian Tugas Akhir
4. Bab 4 Analisa dan Perancangan, menjelaskan tahapan analisa dan perancangan perangkat lunak
5. Bab 5 Implementasi dan Pengujian, menjelaskan hasil implementasi, metode dan hasil pengujian perangkat lunak
6. Bab 6 Penutup, menjelaskan kesimpulan dan saran penelitian.

BAB II

LANDASAN TEORI

2.1 Keamanan Data

Keamanan dan kerahasiaan data saat ini menjadi isu yang sangat penting dan terus berkembang. Beberapa kasus menyangkut keamanan data saat ini menjadi suatu yang membutuhkan biaya penanganan dan pengamanan yang sedemikian besar. Sistem-sistem vital seperti sistem pertahanan, sistem perbankan, dan sistem-sistem setingkat itu membutuhkan tingkat keamanan yang sedemikian tinggi. Hal ini lebih disebabkan karena kemajuan komputer dan konsep *open system*-nya, sehingga siapapun, di manapun dan kapanpun, mempunyai kesempatan untuk mengakses kawasan-kawasan vital tersebut.

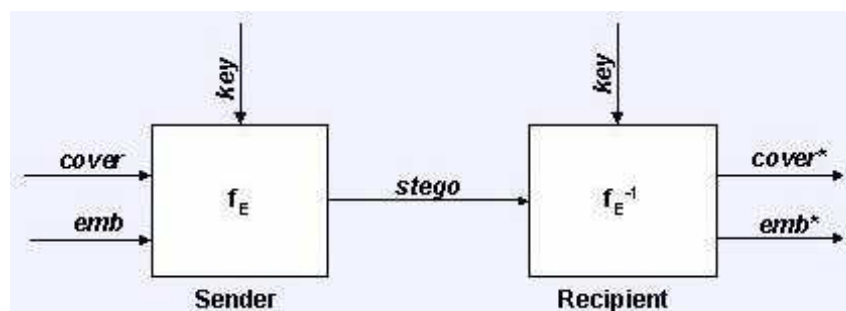
Pada garis besarnya masalah keamanan data dapat dibagi menjadi empat bidang yang saling berhubungan: kerahasiaan, keaslian, pengakuan dan kontrol integritas, yang akan dibahas nanti pada paragraf berikutnya. Kerahasiaan harus dilakukan dengan menjauhkan data dari orang-orang yang tidak berhak. Keaslian berkaitan dengan siapa anda berbicara sebelum memberikan informasi yang sangat penting. Pengakuan berkaitan dengan tanda tangan digital atau sertifikasi digital.

2.2 Steganografi

Kata steganografi (steganography) berasal dari bahasa Yunani *steganos*, yang artinya “tersembunyi atau terselubung”, dan *graphein*, “menulis” sehingga kurang lebih artinya “menulis (tulisan) terselubung”. Teknik ini meliputi banyak sekali metode komunikasi untuk menyembunyikan pesan rahasia.

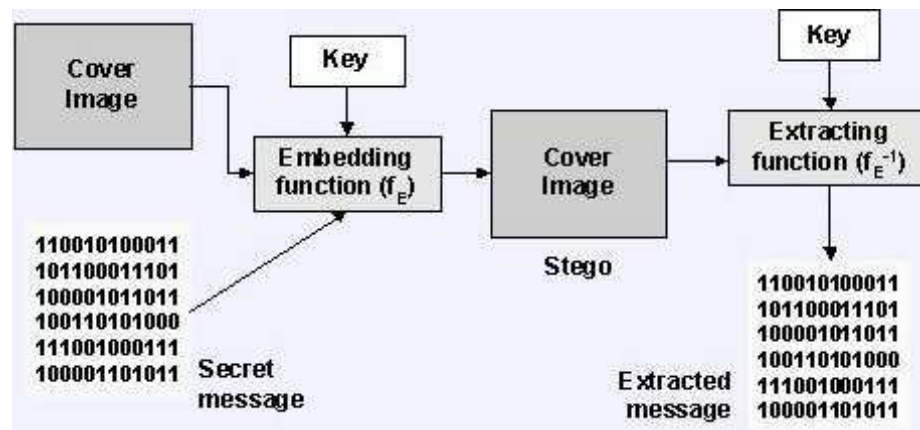
Dalam bidang keamanan komputer, steganography digunakan untuk menyembunyikan data rahasia saat enkripsi tidak dapat dilakukan atau bersamaan

dengan enkripsi. Jadi, walaupun enkripsi berhasil dipecahkan (*decipher*) pesan / data rahasia tetap tidak terlihat. Selain itu, pada *cryptography* pesan disembunyikan dengan “diacak” sehingga pada kasus-kasus tertentu dapat dengan mudah mengundang kecurigaan, sedangkan pada *steganography* pesan “disamarkan” dalam bentuk yang relative “aman” sehingga tidak terjadi kecurigaan itu. *Steganography* dapat digunakan pada berbagai macam bentuk data, yaitu image, audio, dan video.



Gambar 2.1 *Steganographic System*

Gambar 2.1 menunjukkan sebuah sistem steganography umum dimana di bagian pengirim pesan (**sender**), dilakukan proses *embedding* (f_E) pesan yang hendak dikirim secara rahasia (**emb**) ke dalam data cover sebagai tempat menyimpannya (*cover*), dengan menggunakan kunci tertentu (*key*), sehingga dihasilkan data dengan pesan tersembunyi di dalamnya (*stego*). Di bagian penerima pesan (**recipient**), dilakukan proses extracting (f_E^{-1}) pada *stego* untuk memisahkan pesan rahasia (**emb**) dan data penyimpan (*cover*) tadi dengan menggunakan kunci yang sama seperti pada proses embedding tadi. Jadi hanya orang yang tahu kunci ini saja yang dapat mengekstrak pesan rahasia tadi. Proses tadi dapat direpresentasikan secara lebih jelas pada gambar 2 di bawah.



Gambar 2.2 Graphical Version of a Steganographic System

Penyembunyian pesan rahasia ke dalam media penampung pasti mengubah kualitas media tersebut. Kriteria yang harus diperhatikan dalam penyembunyian pesan adalah : (Munir,2004)

1. *Imperceptibility*. Keberadaan pesan rahasia tidak dapat dipersepsi oleh inderawi. Misalnya, jika *coverttext* berupa citra, maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra *coverttext*-nya. Jika *coverttext* berupa audio, maka indera telinga tidak dapat mendeteksi perubahan pada audio *stegotext*-nya.
2. *Fidelity*. Mutu media penampung tidak berubah banyak akibat penyisipan. Perubahan tersebut tidak dapat dipersepsi oleh inderawi. Misalnya, jika *coverttext* berupa citra, maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra *coverttext*-nya. Jika *coverttext* berupa audio, maka audio *stegotext* tidak rusak dan indera telinga tidak dapat mendeteksi perubahan tersebut.
3. *Recovery*. Pesan yang disembunyikan harus diungkapkan kembali (*reveal*). Karena tujuan steganografi adalah *data hiding*, maka sewaktu-waktu pesan rahasia di dalam *stegotext* harus dapat diambil kembali untuk digunakan lebih lanjut.

2.2.1 Metode Steganografi

Kebanyakan algoritma steganografi menggunakan sebuah kombinasi dari bidang jenis teknik untuk melakukan sebuah tugas dalam penyelubungan pesan rahasia dalam sebuah selubung file. Sebuah program steganografi dibutuhkan untuk melakukan hal-hal berikut, baik implisit melalui suatu perkiraan maupun eksplisit melalui sebuah perhitungan: menemukan kelebihan bits dalam selubung file yang dapat digunakan untuk menyelubungi pesan rahasia didalamnya, memilih beberapa diantaranya untuk digunakan dalam menyelubungi data dan penyelubungan data dalam bits dipilih sebelumnya. Berikut ini algoritma atau metode steganografi akan diuji lebih detail.

Ada empat jenis metode Steganography, yaitu :

1. *Least Significant Bit Insertion (LSB)*

LSB merupakan sebuah metode yang lazim digunakan oleh para peneliti pada sebuah steganografi. Hal ini disebabkan karena LSB merupakan sebuah metode steganografi yang paling sederhana, cepat, dan mempunyai kapasitas penyisipan suatu informasi digital yang cukup besar. LSB menyisipkan sebuah informasi rahasia pada bit rendah atau bit yang paling kanan dari sebuah data pixel yang menyusun sebuah informasi digital yang menjadi media penampung suatu informasi rahasia.

2. *Masking and Filtering*

Metode ini biasanya dibatas pada image 24 bit warna dan image *grayscale*. Beberapa literatur menyatakan bahwa metode ini mirip dengan *watermark*, dimana suatu image diberi tanda (*marking*) untuk menyembunyikan pesan rahasia. Hal ini dapat dilakukan dengan memodifikasi *luminance* image dibeberapa bagiannya. Metode ini memiliki *robustness* terhadap kompresi, dan *cropping*. Namun, memiliki batasan kapasitas tertentu pada informasi yang akan disembunyikan.

3. *Algorithm and Transformation*

Metode ini merupakan metode steganografi yang jauh lebih kompleks dari metode-metode sebelumnya, artinya tingkat kesulitan dalam penerapan metode ini lebih tinggi. Untuk menyembunyikan sebuah informasi digital

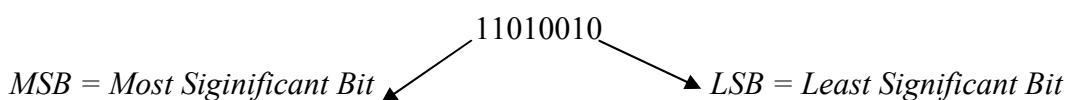
pada media penampungnya dilakukan dengan memanfaatkan *Discrete Cosine Transformation* (DCT) dan *Wavelet Compression*. DCT digunakan pada file-file terkompresi, seperti JPEG yaitu dengan mentransformasikan blok 8 x 8 pixel yang berurutan dari image menjadi 64 koefisien DCT, dan selanjutnya dilakukan penghitungan dengan rumus tertentu. Metode ini sering tidak menimbulkan kecurigaan karena perubahan LSB tidak akan terlihat. Hal ini disebabkan karena metode ini terjadi di domain frekuensi dari sebuah file digital, bukan pada domain spasial, sehingga menyebabkan tidak terlihatnya perubahan pada file digital tersebut.

4. *Spread Spectrum Methode*

Teknik metode ini dalam menyembuyikan suatu informasi digital adalah dengan mengkodekan informasi rahasia dan disebarkan ke setiap spektrum frekuensi yang memungkinkan. Namun, metode ini masih mudah diserang, yaitu dengan cara penghancuran atau pengrusakan dari kompresi dan proses *image* (gambar).

2.3 Least Significant Bit Insertion (LSB)

Penyembunyian data dilakukan dengan mengganti bit-bit data yang tidak terlalu berpengaruh di dalam segmen citra dengan bit-bit data rahasia. Pada susunan bit di dalam sebuah *byte* (1 *byte* = 8 bit), ada bit yang paling berarti (*most significant bit* atau *MSB*) dan bit yang paling kurang berarti (*least significant bit* atau *LSB*). Berikut contoh sebuah susunan bit pada sebuah *byte*:



Bit yang cocok untuk diganti adalah bit *LSB*, sebab perubahan tersebut hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya.

Contohnya pada file image pesan dapat disembunyikan dengan menggunakan cara menyisipkannya pada bit rendah atau bit yang paling kanan (*lsb*) pada data

pixel yang menyusun file tersebut. Jika digunakan image 24 bit color sebagai cover, sebuah bit dari masing-masing komponen Red, Green, dan Blue, dapat digunakan sehingga 3 bit dapat disimpan pada setiap piksel. Sebuah image 800 x 600 piksel dapat digunakan untuk menyembunyikan 1.440.000 bit (180.000 bytes) data rahasia.

Misalnya, di bawah ini terdapat 3 piksel dari image 24 bit color :

```
(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)
```

jika diinginkan untuk menyembunyikan karakter A (10000001) dihasilkan :

```
(00100111 11101000 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)
```

dapat dilihat bahwa hanya 3 bit saja yang perlu diubah untuk menyembunyikan karakter A ini. Perubahan pada LSB ini akan terlalu kecil untuk terdeteksi oleh mata manusia sehingga pesan dapat disembunyikan secara efektif. Jika digunakan image 8 bit color sebagai cover, hanya 1 bit saja dari setiap piksel warna yang dapat dimodifikasi sehingga pemilihan image harus dilakukan dengan sangat hati-hati, karena perubahan LSB dapat menyebabkan terjadinya perubahan warna yang ditampilkan pada citra. Akan lebih baik jika image berupa image grayscale karena perubahan warnanya akan lebih sulit dideteksi oleh mata manusia.

Untuk dapat membuat *hiddentext* tidak dapat dilacak, bit-bit pesan tidak mengganti *byte-byte* yang berurutan, namun dipilih susunan *byte* secara acak. Misalnya jika terdapat 50 *byte* dan 6 bit data yang akan disembunyikan, maka *byte* yang diganti bit *LSB-nya* dipilih secara acak, misalkan *byte* nomor 36, 5, 21, 10, 18, 49. Pembangkitan bilangan acak dilakukan dengan *pseudo-random-number-generator (PRNG)* yang berlaku sebagai kunci stegano. Pada citra 8-bit yang

berukuran 256×256 *pixel* terdapat 65536 *pixel*, setiap *pixel* berukuran 1 *byte* sehingga kita hanya dapat menyisipkan 1 bit pada setiap *pixel*. Pada citra 24-bit yang berukuran 256×256 *pixel*, satu *pixel* berukuran 3 *byte* (atau 1 *byte* untuk setiap komponen R, G, B), sehingga kita bisa menyisipkan pesan sebanyak $65536 \times 3 \text{ bit} = 196608 \text{ bit}$ atau $196608/8 = 24576 \text{ byte}$.

Pesan yang disembunyikan di dalam citra dapat diungkap kembali dengan mengekstraksinya. Posisi *byte* yang menyimpan bit pesan dapat diketahui dari bilangan acak yang dibangkitkan oleh *PRNG*. Jika kunci yang digunakan pada waktu ekstraksi sama dengan kunci pada waktu penyisipan, maka bilangan acak yang dibangkitkan juga sama. Dengan demikian, bit-bit data rahasia yang bertaburan di dalam citra dapat dikumpulkan kembali.

2.4 Citra Digital

Salah satu daya tarik manusia dalam menikmati suatu objek adalah adanya unsur gambar (*image*). Gambar digital merupakan dokumen berbentuk *file* yang dihasilkan melalui perangkat elektronik atau media digital. Gambar digital berupa sekumpulan titik yang disusun dalam bentuk matriks, dan nilainya menyatakan suatu derajat kecerahan (derajat keabuan/*gray-scale*). Derajat keabuan 8 bit menyatakan 256 derajat kecerahan. Pada gambar berwarna nilai setiap titiknya adalah nilai derajat keabuan pada setiap kompoen warna RGB. Bila masing-masing komponen R,G dan B mempunyai 8 bit, maka satu titik dinyatakan dengan $(8+8+8)=24 \text{ bit}$ atau 2^{24} derajat keabuan.

2.4.1 Format Citra Digital

Citra digital merupakan fungsi intensitas cahaya $f(x,y)$, dimana nilai x dan y merupakan koordinat spasial dan harga fungsi tersebut pada setiap titik (x,y) merupakan tingkat kecemerlangan citra pada titik tersebut.

Citra digital adalah citra $f(x,y)$ dimana dilakukan diskritisasi koordinat spasial (*sampling*) dan diskritisasi tingkat kecemerlangannya/keabuan (*kwantisasi*). Citra digital merupakan suatu matriks dimana indeks baris dan kolomnya menyatakan

suatu titik pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar / piksel / pixel / picture element / pels) menyatakan tingkat keabuan pada titik tersebut. Citra digital dinyatakan dengan :

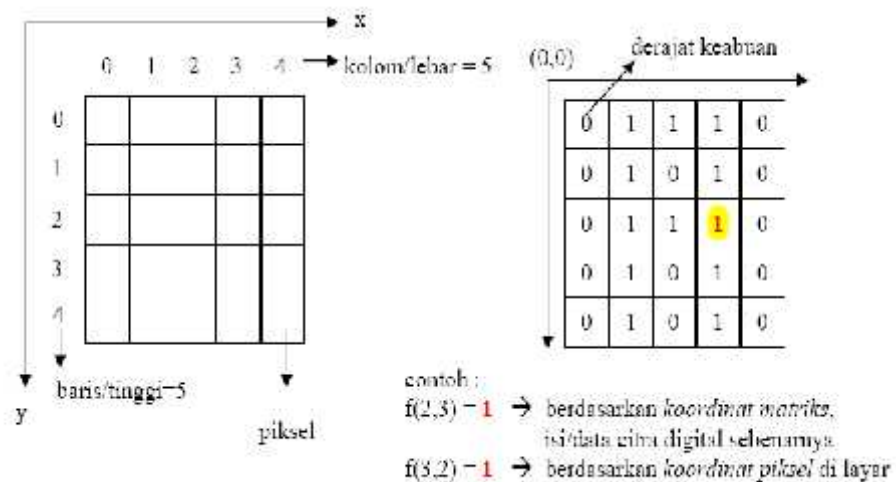
Matriks berukuran $N \times M$ (baris/tinggi = N , kolom/lebar = M)

N = jumlah baris $0 \leq y \leq N - 1$

M = jumlah kolom $0 \leq x \leq M - 1$

L = maksimal warna intensitas (derajat keabuan / gray level) $0 \leq f(x,y) \leq L - 1$

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$



Format Citra

1. Citra digital biasanya berbentuk persegi panjang, *secara visualisasi dimensi ukurannya* dinyatakan sebagai lebar x tinggi
2. Ukurannya dinyatakan dalam titik atau piksel (pixel=picture element)
3. Ukurannya dapat pula dinyatakan dalam satuan panjang (mm atau inci = inch)
4. Resolusi = banyaknya titik untuk setiap satuan panjang (dot per inch).
5. Makin besar resolusi makin banyak titik yang terkandung dalam citra, sehingga menjadi lebih halus dalam visualisasinya.

2.4.2 Citra Bitmap

Struktur *bitmap* secara garis besar dibagi menjadi empat bagian, yaitu *File Header*, *Image Header*, *Color Palette*, dan *Pixel Data*. Perlu diperhatikan, bagian *File Header*, *Image Header*, dan *Color Palette* terdiri atas informasi-informasi yang penting untuk menampilkan citra, apabila terjadi kehilangan data atau kerusakan data pada bagian-bagian ini maka hal tersebut akan mengakibatkan citra rusak atau bahkan tidak bisa ditampilkan.

Struktur file BMP dapat dilihat dari gambar berikut:



Gambar 2.3 Struktur *File BMP*

Pada gambar 2.3 di atas dapat diketahui bahwa *file header* merupakan bagian yang merepresentasikan jenis atau format file atau sebagai bagian identifikasi file. Letaknya di blok pertama yang berfungsi untuk memastikan apakah file ini benar file bitmap atau tidak. Identifikasi ini dapat diketahui dengan menemukan kode bit yang menuliskan BM (bmp). *Image header* merupakan bagian yang berisi informasi ukuran panjang dan lebar file dalam satuan pixel, format warna (jumlah bidang warna / bits-per-pixel), informasi apakah bitmap terkompresi atau tidak serta tipe kompresinya, jumlah data bitmap dalam byte, resolusi, dan jumlah warna yang digunakan. *Color header* merupakan bagian yang berisi informasi intensitas RGB untuk setiap komponen warna pada *pallette*. Dan *pixel data* berisi pixel-pixel data atau gambar.

Pada format *bitmap*, citra disimpan sebagai suatu matriks di mana masing-masing elemennya digunakan untuk menyimpan informasi warna untuk setiap *pixel*. Jumlah warna yang dapat disimpan ditentukan dengan satuan *bit-per-pixel*. Semakin besar ukuran *bit-per-pixel* dari suatu *bitmap*, semakin banyak pula jumlah warna yang dapat disimpan.

Karakteristik lain dari *bitmap* yang juga penting adalah jumlah warna yang dapat disimpan dalam *bitmap* tersebut. Ini ditentukan oleh banyaknya bit yang digunakan untuk menyimpan setiap titik dari *bitmap* yang menggunakan satuan *bpp (bit per pixel)*. Dalam Windows dikenal *bitmap* dengan 1, 4, 8, 16, dan 24 *bit per pixel*. Jumlah warna maksimum yang dapat disimpan dalam suatu *bitmap* adalah sebanyak 2^n , dimana n adalah banyaknya bit yang digunakan untuk menyimpan satu titik dari *bitmap*.

Agar *stego-image* dapat ditampilkan persis dengan aslinya, dalam melakukan steganografi, yang disisipi pesan hanya bagian *pixel data* saja karena jika bagian *file header*, *image header*, dan *color palette* ikut disisipi pesan, maka bagian citra tidak dapat ditampilkan lagi. Sebagai contoh, salah satu bagian dari *file header* adalah *bfType* yang mengandung karakter “BM” yang mengidentifikasi tipe arsip, apabila tipe arsip ini disisipi pesan, maka tipe arsip dapat berubah menjadi tidak dikenali sehingga citra tidak dapat ditampilkan. Hal ini menunjukkan bahwa penyisipan pesan dengan teknik *LSB* hanya dapat dilakukan pada bagian *pixel data*, agar citra yang menyembunyikan tidak rusak.

Pada representasi arsip 24-bit *bitmap*, setiap piksel akan terdiri dari 3 byte karena setiap 1 byte akan merepresentasikan nilai red, blue, atau green. Apabila terdapat pemilihan nilai *LSB* tertentu akan dibagi secara merata pada 3 representasi warna tersebut.

2.5 Steganografi Pada Media Digital File Gambar

Pada komputer, gambar yang tampil di layar monitor merupakan kumpulan array yang merepresentasikan intensitas cahaya yang bervariasi pada pixel. Pixel adalah titik dilayar monitor yang dapat diatur untuk menampilkan warna tertentu. Pixel disusun dilayar monitor dalam susunan baris dan kolom. Susunan ini disebut

resolusi monitor. Resolusi monitor yang sering dijumpai adalah 640x480, 800x600, 1024x768. Resolusi 640x480 akan menampilkan pixel sejumlah 640 baris dan 480 kolom, sehingga total pixel yang digunakan adalah $640 \times 480 = 307.200$ pixel. Melalui pixel inilah, suatu gambar dapat dimanipulasi untuk menyimpan informasi yang akan digunakan sebagai salah satu pengimplementasian steganografi.

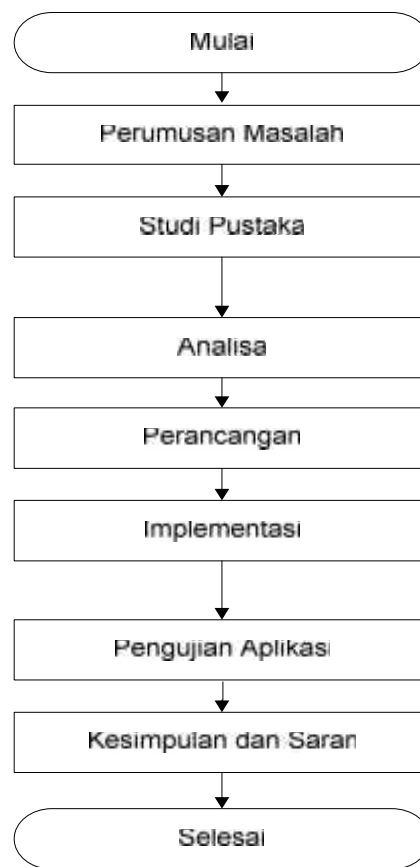
Untuk dapat membuat *hiddentext* tidak dapat dilacak, bit-bit pesan tidak mengganti *byte-byte* yang berurutan, namun dipilih susunan *byte* secara acak. Misalnya jika terdapat 50 *byte* dan 6 bit data yang akan disembunyikan, maka *byte* yang diganti bit *LSB*-nya dipilih secara acak, misalkan *byte* nomor 36, 5, 21, 10, 18, 49. Pembangkitan bilangan acak dilakukan dengan *pseudo-random-number-generator* (*PRNG*) yang berlaku sebagai kunci stegano. Pada citra 24-bit yang berukuran 256×256 *pixel*, satu *pixel* berukuran 3 *byte* (atau 1 *byte* untuk setiap komponen R, G, B), sehingga kita bisa menyisipkan pesan sebanyak $65536 \times 3 \text{ bit} = 196608 \text{ bit}$ atau $196608/8 = 24576$ *byte*.

Steganografi pada media penampung file gambar digunakan untuk mengeksploitasi keterbatasan kekuatan system penglihatan manusia dengan cara menurunkan kualitas warna pada media penampung file gambar yang belum disisipi pesan rahasia. Sehingga dengan keterbatasan tersebut manusia sulit menemukan gradasi penurunan kualitas warna pada media penampung file gambar yang telah disisipi file rahasia.

BAB III

METODOLOGI PENELITIAN

Metodologi penelitian menguraikan tahapan seluruh kegiatan yang dilaksanakan selama kegiatan penelitian berlangsung. Adapun langkah-langkah yang dilalui dalam pelaksanaan penelitian ini adalah sebagai berikut:



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1 Perumusan Masalah

Dengan memanfaatkan informasi-informasi yang didapat dari penelitian pendahuluan yang telah dilakukan, maka dilakukan tahap berikutnya yaitu mengidentifikasi masalah. Pada tugas akhir ini masalah yang akan diidentifikasi

adalah melakukan studi analisis metode *Least Significant Bit Insertion* (LSB) untuk diimplementasikan dalam penyandian citra digital bertipe bitmap.

3.2 Studi Pustaka

Tahap ini merupakan langkah yang dilakukan untuk mendapatkan teori-teori lanjutan yang dibutuhkan, yaitu dengan melakukan Studi Pustaka (*Library Research*). Studi pustaka ini bertujuan untuk mendapatkan dasar-dasar pengetahuan yang akan diterapkan dalam penelitian dan memperoleh informasi dalam tahap persiapan penelitian ini, maka dipelajari bahan pustaka yang ada kaitannya dengan penelitian metode *Least Significant Bit Insertion* (LSB), dan penyandian citra.

3.3 Analisa

Tahap ini merupakan tahap analisa terhadap algoritma yang digunakan berdasarkan teori-teori yang berkaitan. Tahap ini meliputi analisa terhadap metode *Least Significant Bit Insertion* (LSB).

3.4 Perancangan Program

Pada tahap ini, dilakukan perancangan terhadap perangkat lunak yang akan dibangun. Perancangan ini meliputi perancangan struktur menu dan perancangan *interface*.

3.5 Implementasi

Tahap implementasi merupakan tahap penerjemahan hasil analisa ke dalam bentuk *coding* sesuai dengan hasil perancangan yang telah dibuat.

3.6 Pengujian

Selanjutnya dilakukan pengujian terhadap aplikasi yang telah dibangun agar dapat diketahui hasilnya. Jika terdapat *error*, maka proses akan kembali ke tahap analisis untuk dilakukan analisa ulang.

3.7 Kesimpulan dan saran

Berdasarkan hasil pengujian dihasilkan kesimpulan yang sesuai dengan rumusan masalah dan tujuan yang akan dicapai, serta saran-saran yang diperlukan untuk pengembangan selanjutnya

BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisis

Analisis perangkat lunak dalam mengembangkan aplikasi steganografi *file* pada *citra bitmap* dengan metode *Least Significant Bit* (LSB) meliputi analisis umum, analisis rinci, sehingga pengembangan aplikasi sesuai dengan maksud dan tujuan yang ingin dicapai dalam penelitian ini.

4.1.1 Analisis Umum

Keamanan data dan informasi merupakan hal yang sangat penting dalam sistem jaringan komputer. Seiring dengan perkembangan teknologi informasi, semakin berkembang pula teknik kejahatan yang berupa perusakan maupun pencurian data oleh pihak yang tidak memiliki wewenang atas data tersebut. Ada beberapa bentuk penyerangan terhadap data dan informasi, seperti *hacker*, *cracker*, *trojan force attack*, dan lain-lain. Oleh karena itu, pada saat ini telah dilakukan berbagai upaya untuk menjaga keamanan data dan mengatasi serangan-serangan tersebut.

Salah satu cara dalam mengimplementasikan keamanan data dan informasi dalam sistem jaringan komputer adalah dengan teknik steganografi. Steganografi merupakan salah satu cara untuk menyembunyikan suatu pesan atau data rahasia pada suatu media yang tampak tidak mengandung apa-apa, kecuali bagi orang yang mengerti kuncinya. Sebelumnya telah dilakukan teknik steganografi pada *file image* menggunakan metode *Least Significant Bit* yang menghasilkan perubahan yang tidak berarti pada gambar karena penyembunyian data dilakukan dengan mengganti beberapa *bit data* di dalam segmen citra dengan *bit-bit data* rahasia. Pada susunan *bit* di dalam sebuah *byte* ($1 \text{ byte} = 8 \text{ bit}$), ada *bit* yang paling berarti (*Most Significant Bit*) dan *bit* yang paling kurang berarti (*Least*

Significant Bit). Oleh karena itu dilakukan pengembangan metode *Least Significant Bit* untuk melihat perubahan yang akan terjadi apabila penyembunyian data dilakukan pada 3 bit terendah. Kemudian hasil keluaran (*output*) yang berupa gambar yang telah disisipkan *file* akan diukur kualitas gambarnya.

4.1.2 Analisis Rinci

Didalam proses ini terdapat dua file biner dimana file yang pertama adalah file biner hasil dari konvers file gambar kedalam biner sedangkan file yang kedua adalah file biner yang akan disisipkan dikonvers kedalam file biner.

Konsep dari penyisipan ini adalah bahwa setiap delapan bit file pertama akan disisipkan satu bit file yang berasal dari file kedua. Berikut ini gambaran dari proses tersebut.

File Pertama gambar bitmap (contoh simulasi file biner hasil konvers dari gambar ke biner)

```
11000011 11010011 11000011 10100110 00011101 01111000 01110111
01000001 10111011 01001110 11001100 00111010 01100000 11101101
10100111 00001100 00011100 11111010 11110000 11100001 10110011
11001100 00011011 01110000 11101110 11000011 10011111 00101110
11011100 10111011 10111010 01000001 11001111 10100111 00101100
00111101 00110010 11100111 11010011 11001110 01100110 11111110
11011001 01100000 10110011 10111111 10101111 00101000 00101011
10011010 10110011 01011111 00011011 11100101 11100101 01110010
```

File Kedua data.doc (contoh simulasi file biner hasil konvers *file* yang akan disisipkan ke file biner)

```
11100111 11000011 01001110 00011001 00011100 11001110 01100110
```

File hasil penyisipan di bit 8 (biner yang ber garis bawah adalah hasil biner yang disipkan dari file kedua)

```
11000111 11010101 11000011 10100100 00011010 01111011 01110111
01000101 10111111 01001011 11001110 00111100 01100110 11101010
10100111 00001101 00011110 11111101 11110000 11100010 10110101
11001101 00011101 01110000 11101100 11000110 10011010 00101011
11011111 10111110 10111010 01000001 11001010 10100100 00101110
```

0011111 <u>1</u>	0011001 <u>1</u>	1110011 <u>1</u>	1101000 <u>0</u>	1100110 <u>0</u>	0110010 <u>1</u>	1111110 <u>1</u>
1101101 <u>0</u>	0110001 <u>0</u>	1011001 <u>1</u>	1011110 <u>1</u>	1010110 <u>1</u>	0010100 <u>0</u>	0010100 <u>0</u>
1001101 <u>1</u>	1011001 <u>1</u>	0101111 <u>0</u>	0001101 <u>0</u>	1110010 <u>1</u>	1110010 <u>1</u>	0111001 <u>0</u>

File hasil penyisipan di bit 7 (biner yang ber garis bawah adalah hasil biner yang disipkan dari file kedua)

1100001 <u>1</u>	1101001 <u>1</u>	1100001 <u>1</u>	1010010 <u>0</u>	0001110 <u>1</u>	0111101 <u>0</u>	0111011 <u>1</u>
0100001 <u>1</u>	1011101 <u>1</u>	0100111 <u>0</u>	1100110 <u>0</u>	0011100 <u>0</u>	0110000 <u>0</u>	1110110 <u>1</u>
1010011 <u>1</u>	0000111 <u>0</u>	0001110 <u>0</u>	1111101 <u>0</u>	1111000 <u>0</u>	1110000 <u>1</u>	1011001 <u>1</u>
1100111 <u>0</u>	0001101 <u>1</u>	0111000 <u>0</u>	1110110 <u>0</u>	1100000 <u>1</u>	1001110 <u>1</u>	0010111 <u>0</u>
1101111 <u>0</u>	1011100 <u>1</u>	1011100 <u>0</u>	0100001 <u>1</u>	1100110 <u>1</u>	1010011 <u>1</u>	0010111 <u>0</u>
0011111 <u>1</u>	0011001 <u>0</u>	1110011 <u>1</u>	1101000 <u>1</u>	1100110 <u>0</u>	0110011 <u>0</u>	1111111 <u>0</u>
1101100 <u>1</u>	0110000 <u>0</u>	1011001 <u>1</u>	1011111 <u>1</u>	1010111 <u>1</u>	0010100 <u>0</u>	0010100 <u>1</u>
1001101 <u>0</u>	1011001 <u>1</u>	0101110 <u>1</u>	0001100 <u>1</u>	1110011 <u>1</u>	1110011 <u>1</u>	0111000 <u>0</u>

File hasil penyisipan di bit 6 (biner yang ber garis bawah adalah hasil biner yang disipkan dari file kedua)

1100011 <u>1</u>	1101011 <u>1</u>	1100011 <u>1</u>	1010001 <u>0</u>	0001100 <u>1</u>	0111110 <u>0</u>	0111011 <u>1</u>
0100010 <u>1</u>	1011111 <u>1</u>	0100111 <u>0</u>	1100101 <u>0</u>	0011100 <u>0</u>	0110001 <u>0</u>	1110100 <u>1</u>
1010011 <u>1</u>	0000110 <u>0</u>	0001101 <u>0</u>	1111110 <u>0</u>	1111000 <u>0</u>	1110001 <u>1</u>	1011010 <u>1</u>
1100110 <u>0</u>	0001110 <u>1</u>	0111000 <u>0</u>	1110100 <u>0</u>	1100001 <u>1</u>	1001101 <u>1</u>	0010111 <u>0</u>
1101110 <u>0</u>	1011101 <u>1</u>	1011101 <u>0</u>	0100010 <u>1</u>	1100101 <u>1</u>	1010001 <u>1</u>	0010101 <u>0</u>
0011111 <u>1</u>	0011011 <u>0</u>	1110011 <u>1</u>	1101001 <u>1</u>	1100101 <u>0</u>	0110011 <u>0</u>	1111111 <u>0</u>
1101100 <u>1</u>	0110000 <u>0</u>	1011011 <u>1</u>	1011110 <u>1</u>	1010110 <u>1</u>	0010100 <u>0</u>	0010101 <u>0</u>
1001111 <u>0</u>	1011011 <u>1</u>	0101101 <u>1</u>	0001101 <u>1</u>	1110010 <u>1</u>	1110010 <u>1</u>	0111001 <u>0</u>

Berikut ini algoritma penyisipan serta penjelasan:

Algoritma 3.1 Algoritma penyisipan

algoritma untuk menyisipkan biner file kedalam biner gambar

Procedure Penyisipan;

deklarasi:

```
1.Temp_Rep_File_Kedua   :string
2.File_Kedua            :integer
3.Jml_Rep_File_Kedua    :string
4.Temp_File_Kesatu      :string
5.I                     :integer
6.Hasil_Sisip           :string
```

Algoritma:

```
{menghilangkan tanda spasi pemisah bit dalam file kedua}
7.Temp_Rep_File_Kedua <- Replace(File_Kedua, " ", "")
{ menghitung jumlah file }
8.Jml_Rep_File_Kedua <- Len(Temp_Rep_File_Kedua)
{mengambil file pada file dari kiri sebanyak ((Jml * 8) + Jml)}
9.Temp_File_Kesatu <- Left(File_Kesatu, (Jml * 8) + Jml)
{ menghilangkan tanda spasi pemisah bit dalam file Temp_File_Kesatu}
10.Temp_Rep_File_Kesatu <- Replace(Temp_File_Kesatu, " ", "")
{ inisial No dan Nil}

11. No ← 1
12. Nil ← 1
{Perulangan dari 1 sampai dengan Jml_Rep_File_Kedua }
13. For i ← 1 To Jml_Rep_File_Kedua do
    { jika hasil sisip masih kosong}
14.   If Hasil_Sisip ← "" Then
        { pernyisipan 8 bit pertama}
15.     Hasil_Sisip ← Hasil_Sisip & "" & Mid(Temp_
Rep_File_Kesatu,
        i, ((i * 8) - 1)) & Mid(Temp_Rep_File_Kedua, i, 1) & "
    "
    Else
        A ← Mid(Temp_Rep_File_Kesatu, ((i * 8) - 7), 7)
```

4.1.3 Analisis Kebutuhan Sistem

4.1.3.1 Analisis Data

Dalam mengembangkan aplikasi steganografi pada *gambar* dengan menggunakan metode *Least Significant Bit* (LSB) ini diperlukan data-data agar aplikasi dapat berjalan sesuai dengan apa yang diinginkan, data-data yang dibutuhkan untuk implementasi aplikasi adalah sebagai berikut:

1. Data File

Data *file* yang akan disisipkan ke dalam gambar.

2. Media Penampung

Media penampung, yaitu media yang berupa *gambar* dengan *format* .bmp sebagai penampung yang akan disisipkan file.

4.1.3.2 Analisis Masukan

Didalam aplikasi steganografi pada gambar ini, masukan (*input*) aplikasi dibagi menjadi 3, yaitu:

1. *Input* file yang akan disisipkan ke dalam *gambar*.
2. *Input* gambar penampung data file yang akan di-steganografi.

4.1.3.3 Analisis Proses

Setelah masukan terhadap aplikasi dilakukan, ada beberapa proses di dalam aplikasi ini, yaitu:

1. Manipulasi data.

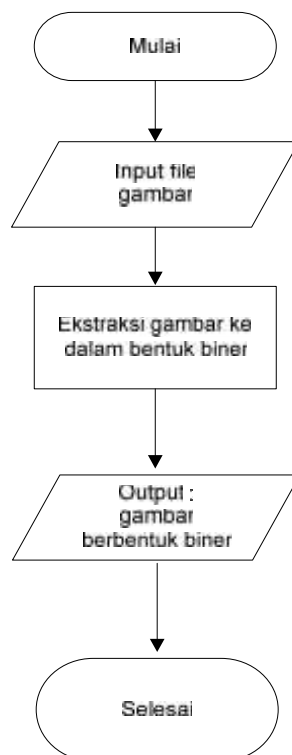
Proses memasukkan *file* ke dalam aplikasi. Dalam proses ini *file* yang akan disisipkan diubah ke bentuk kode ASCII yang selanjutnya diubah ke bentuk biner.

2. Proses pemilihan gambar sebagai citra penampung.

Merupakan proses pemilihan *file* gambar sebagai media penampung *file* yang akan disisipkan.

3. Proses ekstraksi data gambar ke biner.

Sebelum *file* disisipkan, *file* gambar harus diekstraksi ke dalam bentuk biner. Proses ekstraksi gambar ke bentuk biner melalui beberapa tahap yang dapat dilihat pada gambar berikut:

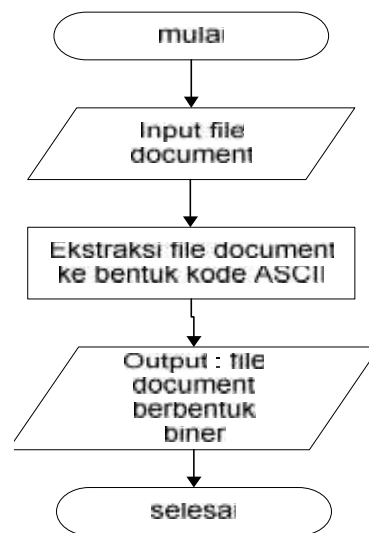


Gambar 4.1. Flowchart Ekstraksi Gambar Ke Bentuk Biner

4. Proses ekstraksi *file* ke bentuk biner.

- a) Pada proses ini terlebih dahulu *file* yang diinputkan diekstraksikan ke bentuk kode ASCII.
- b) Setelah itu, dihasilkan *file* yang sudah berbentuk biner.

Proses ekstraksi *file* ke bentuk biner melalui beberapa tahap yang dapat dilihat pada gambar berikut:



Gambar 4.2. Flowchart Ekstraksi File Ke Bentuk Biner

5. Proses penyisipan *file* ke dalam gambar.

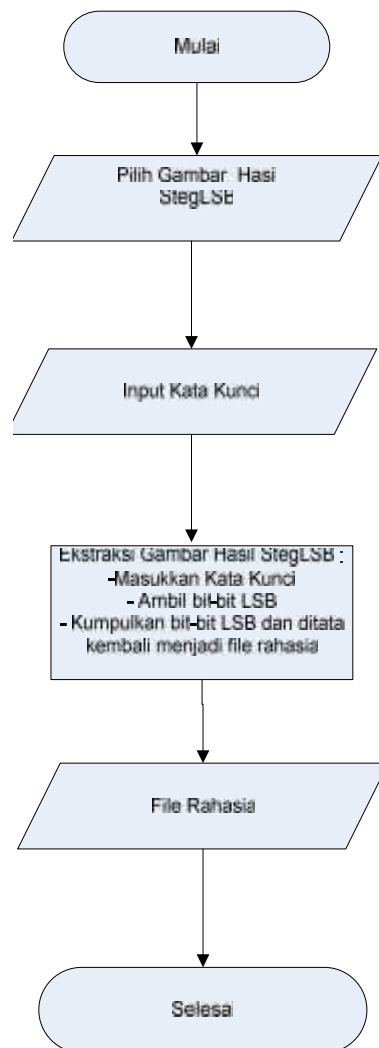
Proses penyisipan *file* ke dalam gambar dilakukan dengan menggunakan metode *least significant bit* (LSB). Sebagai contoh citra penampung yang telah diubah menjadi kode biner adalah **10011000 10111000 00110111 01001110** dan data rahasia yang akan disisipkan sebelumnya telah diubah ke kode biner adalah **1100** maka hasil steganografi menggunakan metode LSB menghasilkan **10011001 10111001 00110110 01001110**. Flowchart pada aplikasi ini menggambarkan urutan proses aplikasi steganografi berupa urutan proses *input* gambar, ekstraksi gambar ke biner, *input file*, ekstraksi

file ke bentuk biner, urutan proses penyisipan gambar, serta proses ekstrak data.



Gambar 4.3. Flowchart Proses Penyisipan File Kedalam Gambar

6. Proses *input* kata kunci (*password*) untuk membuka (ekstrak) *file* dari gambar. Kata kunci yang digunakan merupakan kata kunci yang diberikan aplikasi secara acak pada saat proses penyisipan file ke dalam gambar.



Gambar 4.4. Flowchart Proses Ekstraksi Hasil

4.1.3.4 Analisis Keluaran Gambar Yang Disisipkan File

Setelah proses steganografi dilakukan, aplikasi ini akan menampilkan sebuah keluaran (*output*) yang sesuai dengan masukan (*input*) pada saat steganografi dilakukan, keluaran dari aplikasi steganografi ini berupa *file* yang sebelumnya disisipkan pada gambar *digital*.

Kemudian kualitas hasil steganografi gambar yang telah disisipkan file akan diukur dengan menggunakan rumus PSNR (*peak signal – to – noise ratio*).

Untuk menentukan PSNR, terlebih dahulu harus ditentukan nilai rata-rata kuadrat dari error (MSE - *Mean Square Error*). Perhitungan MSE adalah sebagai berikut :

$$MSE = \frac{1}{mn} \sum_i^m \sum_j^n \|I(i, j) - K(i, j)\|^2 \dots\dots\dots(IV-1)$$

Dimana :

MSE = Nilai *Mean Square Error* dari citra tersebut

m = panjang citra tersebut (dalam piksel)

n = lebar citra tersebut (dalam piksel)

(i,j) = koordinat masing-masing piksel

I = citra digital yang asli sebelum distegano

K = citra digital sesudah distegano

e = nilai error pada gambar

Sementara nilai PSNR dihitung dari kuadrat nilai maksimum sinyal dibagi dengan MSE. Apabila diinginkan PSNR dalam desibel, maka nilai PSNR akan menjadi sebagai berikut :

$$PSNR = 10 \cdot \log \left(\frac{MAX_i^2}{MSE} \right) = 20 \cdot \log \left(\frac{MAX_i}{\sqrt{MSE}} \right) \dots\dots(IV-2)$$

Dimana :

PSNR = nilai PSNR citra (dalam dB)

MAX_i = nilai maksimum piksel

MSE = nilai MSE

db = satuan unit logaritmik yang digunakan untuk mendeskripsikan rasio gambar

Peak Signal-to-Noise Ratio (PSNR) sangat berkaitan erat dengan MSE. Hubungan antara MSE dan PSNR berbanding terbalik. Semakin kecil nilai MSE berarti nilai eror semakin kecil. Semakin tinggi nilai PSNR berarti semakin bagus karena rasio *Signal-to-Noise* akan semakin tinggi dan tidak banyak data yang mengalami perubahan. Pada umumnya, pemrosesan gambar dapat diterima oleh mata manusia jika nilai PSNR lebih besar dari 30 dB. Semakin besar PSNR, maka semakin baik kualitas gambar yang dihasilkan.

Algoritma PSNR yang digunakan dibuat menggunakan bahasa pemrograman matlab mengingat matlab merupakan bahasa pemrograman sangat baik untuk mengolah *file* citra karena dilengkapi fungsi-fungsi yang memudahkan pemakaiannya. Dibawah ini merupakan program yang digunakan untuk mengetahui nilai PSNR dari setiap file citra sebelum dan sesudah disisipkan pesan.

4.2 Perancangan

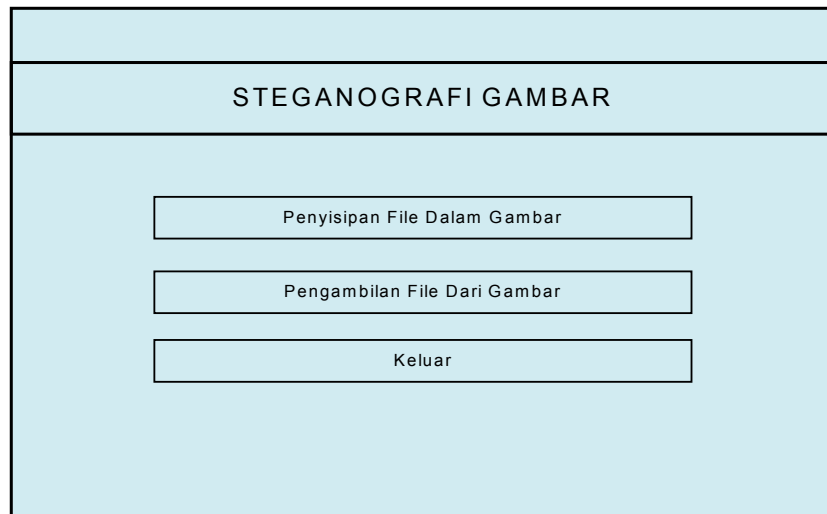
Perancangan perangkat lunak dalam membangun aplikasi steganografi *file* pada *citra digital* dengan metode *Least Significant Bit* (LSB) meliputi perancangan antarmuka proses steganografi dan perancangan antarmuka proses ekstraksi hasil.

4.2.1. Perancangan Antarmuka Aplikasi

Antarmuka atau *interface* merupakan suatu sarana yang memungkinkan terjadinya interaksi antara manusia dan komputer. Oleh sebab itu, *interface* dari sebuah perangkat lunak yang akan dibangun harus bersifat *user friendly* yang bertujuan agar pengguna (*user*) dapat mengerti dengan mudah dan memahami cara menggunakan perangkat lunak ini.

4.2.1.1. Perancangan Antarmuka Proses Penyisipan File

Gambaran antarmuka (*interface*) yang dibutuhkan aplikasi yang akan dibangun ini terdiri dari beberapa *form* yang diakses *user*. Salah satu perancangan tampilan utama aplikasi steganografi ini adalah sebagai berikut :



Gambar 4.5 Rancangan Menu Utama

Keterangan rancangan menu utama dapat dilihat pada tabel berikut :

Table 4.1 Tabel Keterangan Rancangan Menu Utama

Objek	Properti	Pengaturan
Label 1	Caption	STEGANOGRAFI GAMBAR
Command Button1	Caption	Penyisipan File Dalam Gambar
Command Button2	Caption	Pengambilan File Dari Gambar
Command Button3	Caption	Keluar

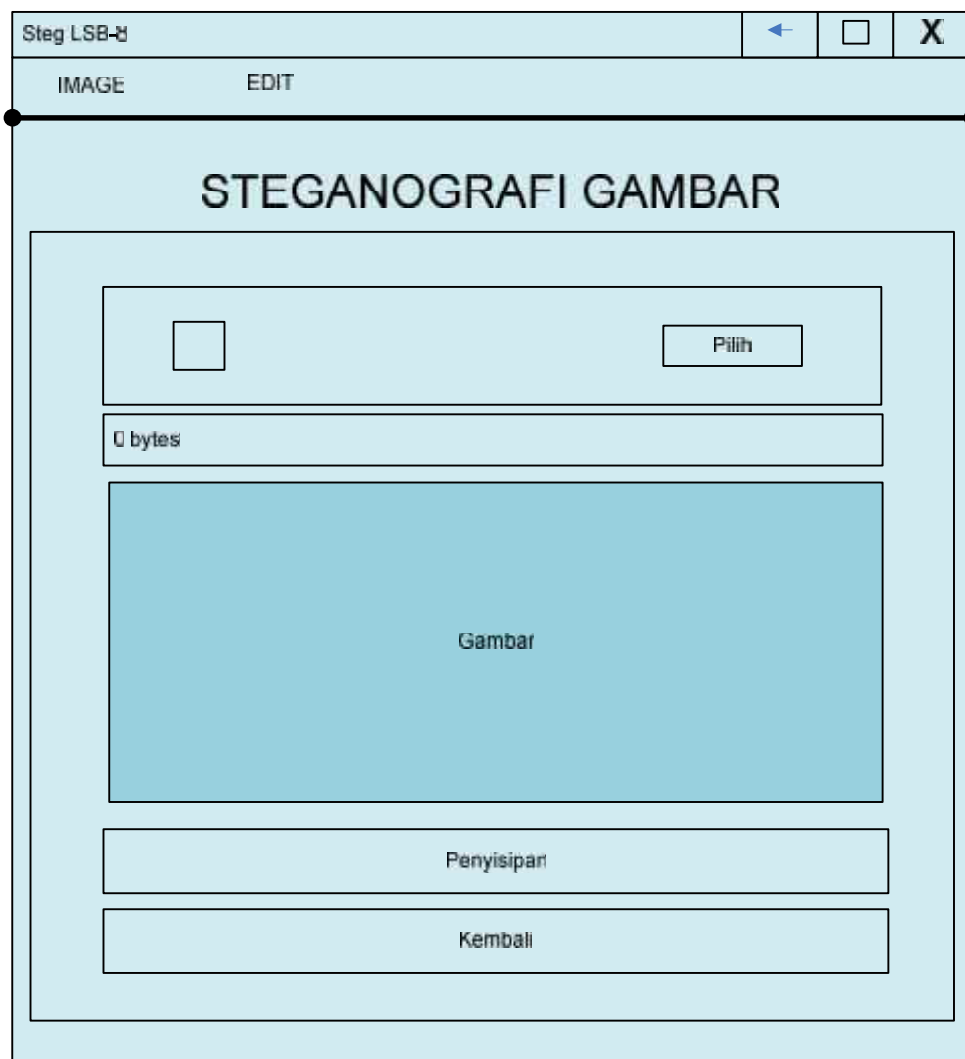
Perancangan berikut ini merupakan rancangan antar muka proses pemilihan bit untuk penyisipan *file*. Pada *system* ini *user* dapat memilih bit gambar yang akan disisipkan *file*, yaitu bit ke 8,7,6, dan 5 yang telah disediakan di *combo box*. Kemudian menekan tombol “ Pilih “ untuk melakukan proses penyisipan.

Gambar 4.6 Rancangan Menu Stegano LSB

Keterangan rancangan menu stegano LSB pilihan dapat dilihat pada table berikut :

Objek	Properti	Pengaturan
Label1	Caption	STEGLSB
Option Button 1	Caption	Penyisipan Pada Bit 8
Option Button 2	Caption	Penyisipan Pada Bit 7
Option Button 3	Caption	Penyisipan Pada Bit 6
CommandButton1	Caption	Pilih

Perancangan berikut ini merupakan rancangan proses penyisipan *file* berdasarkan bit gambar yang sudah dipilih di *form* sebelumnya. Untuk melakukan prosesnya, *user* dapat menekan tombol “Pilih” untuk memilih *file* yang akan disisipkan dan memilih gambar sebagai media pembawa pesan *file*.



Gambar 4.7 Rancangan Menu Penyisipan File

Keterangan rancangan menu penyisipan file pada tabel berikut ini:

Objek	Properti	Pengaturan
Menu Editor	Caption	Image, Edit
Image1	ImgIcon	Image
Image2	ImgPreview	Image
Picture Box1	PicStatus	PicStatus
Picture Box2	PicInfo	PicInfo
CommandButton1	Caption	Pilih

CommandButton2	Caption	Penyisipan
CommandButton3	Caption	Kembali

Berikut ini *form* kata kunci dalam proses penyisipan file ke gambar.

Gambar 4.8 Rancangan Kata Kunci Penyisipan

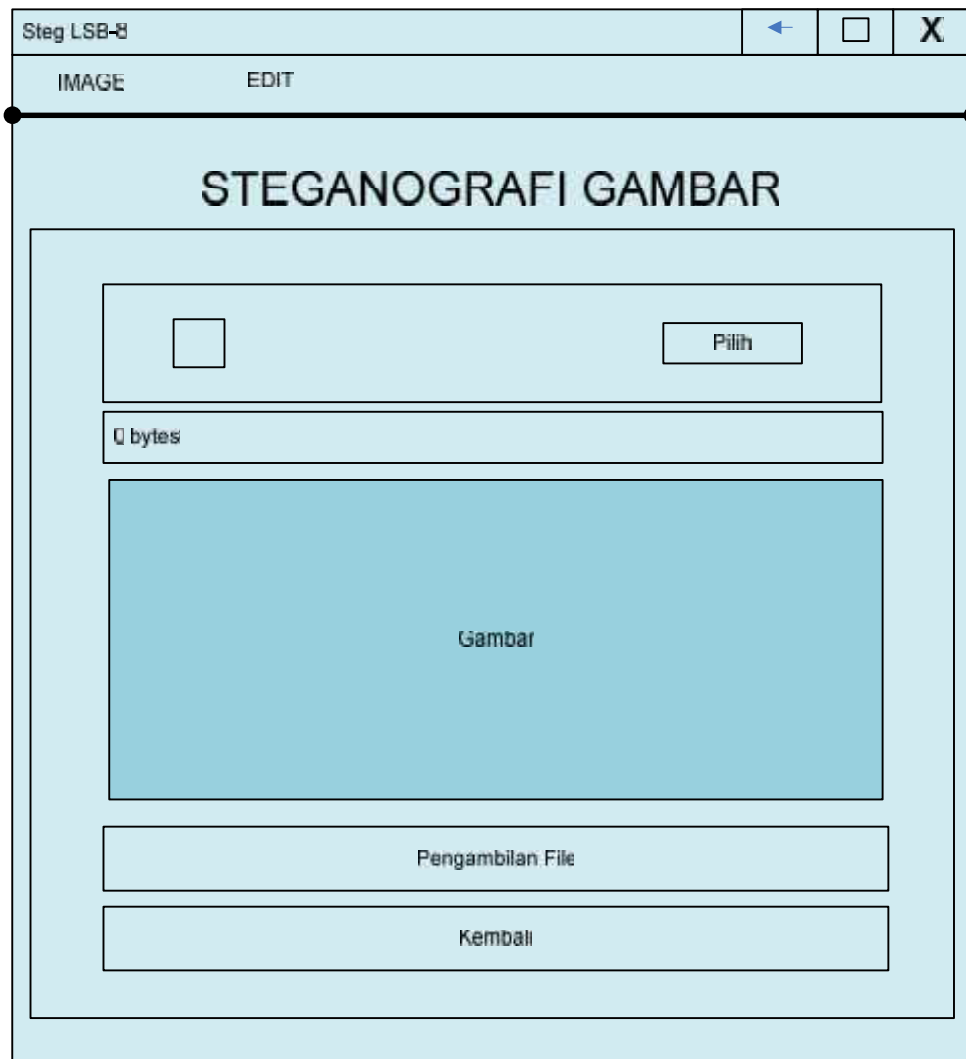
Keterangan rancangan kata kunci penyisipan pada tabel berikut ini:

Objek	Properti	Pengaturan
Label1	Caption	Penyisipan File ke Gambar
Label2	Caption	Kata kunci
Label3	Caption	Kualitas kunci
Label4	Caption	Ulang kata kunci
TextBox1	Text	Kata kunci
TextBox2	Text	Ulang kunci kata
CommandButton1	Caption	Sisip
CommandButton2	Caption	Tunda

4.2.1.2 Perancangan Antarmuka Proses Ekstraksi

Perancangan antarmuka proses ekstraksi meliputi beberapa *form* yang harus dijalankan oleh *user*. Berikut ini merupakan antarmuka menu pilih gambar untuk melakukan ekstraksi gambar hasil steganografi. Untuk melakukan proses ekstraksi *user* harus menekan pilihan *radio button* “**Pengambilan File dari**

Gambar” yang terdapat pada menu utama (gambar 4.5). Apabila diklik akan muncul tampilan sebagai berikut:



Gambar 4.9 Rancangan Pengambilan File

Keterangan rancangan menu penyisipan file pada tabel berikut ini:

Objek	Properti	Pengaturan
Menu Editor	Caption	Image, Edit
Image1	ImgIcon	Image
Image2	ImgPreview	Image

Picture Box1	PicStatus	PicStatus
Picture Box2	PicInfo	PicInfo
CommandButton1	Caption	Pilih
CommandButton2	Caption	Pengambilan File
CommandButton3	Caption	Kembali

Berikut ini *form* kata kunci dalam proses pengambilan *file* ke gambar.

Gambar 4.10 Rancangan Kata Kunci Pengambilan File

Keterangan rancangan kata kunci pengambilan file pada tabel berikut ini:

Objek	Properti	Pengaturan
Label1	Caption	Pengambilan File dari Gambar
Label2	Caption	Kata kunci
Textbox	Text	Kata kunci
CommandButton1	Caption	Ok
CommandButton2	Caption	Keluar

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi Sistem

Implementasi merupakan lanjutan dari tahap perancangan yaitu aplikasi siap dioperasikan pada keadaan yang sebenarnya, sehingga akan diketahui apakah aplikasi yang dibuat telah menghasilkan tujuan yang diinginkan.

Program aplikasi steganografi file pada citra bitmap dengan menerapkan metode *least significant bit (LSB)* memanfaatkan perangkat lunak *Microsoft Visual Basic 6.0*.

5.1.1 Alasan Pemilihan Perangkat Lunak

Perangkat lunak yang digunakan dalam implementasi steganografi ini adalah *Microsoft Visual Basic 6.0* untuk penanganan antar mukanya berdasarkan beberapa pertimbangan yaitu:

Microsoft Visual Basic 6.0 hampir dapat memanfaatkan seluruh kemudahan dan kecanggihan yang dimiliki oleh sistem operasi *Windows*. Apalagi dengan adanya *Object Oriented Programming (OOP)*, objek-objek yang disediakan mudah digunakan sehingga dapat dibuat aplikasi yang sesuai dengan tampilan dan cara kerja *Windows*.

5.1.2 Alasan Pemilihan *File BMP*

Pada dasarnya metode *least significant bit (LSB)* yang dirancang dapat menggunakan semua *file*, namun pada tugas akhir ini penulis hanya membatasi *file bmp* sebagai media pembawa karena *file bmp* merupakan *file bmp* memiliki ukuran yang jauh lebih besar dari pada tipe-tipe yang lain. Selain itu *file bmp* banyak digunakan dan hampir dapat dibuka disemua program pengolah gambar,

file bmp juga mudah untuk disandikan karena ukuran *byte*-nya dapat dihitung tinggi dan lebarnya.

Pada format *bitmap*, citra disimpan sebagai suatu matriks di mana masing-masing elemennya digunakan untuk menyimpan informasi warna untuk setiap *pixel*. Jumlah warna yang dapat disimpan ditentukan dengan satuan *bit-per-pixel*. Semakin besar ukuran *bit-per-pixel* dari suatu *bitmap*, semakin banyak pula jumlah warna yang dapat disimpan.

Karakteristik lain dari *bitmap* yang juga penting adalah jumlah warna yang dapat disimpan dalam *bitmap* tersebut. Ini ditentukan oleh banyaknya bit yang digunakan untuk menyimpan setiap titik dari *bitmap* yang menggunakan satuan *bpp* (*bit per pixel*). Dalam Windows dikenal *bitmap* dengan 1, 4, 8, 16, dan 24 *bit per pixel*. Jumlah warna maksimum yang dapat disimpan dalam suatu *bitmap* adalah sebanyak 2^n , dimana n adalah banyaknya bit yang digunakan untuk menyimpan satu titik dari *bitmap* (Taufik, 2008).

5.1.3 Batasan Implementasi

Batasan implementasi pada penulisan tugas akhir ini adalah

1. Aplikasi ini tidak menyimpan kunci proses penyisipan, sehingga apabila *user* lupa atau kehilangan kunci maka tidak dapat mengambil kembali data yang telah disisipkan.
2. Aplikasi ini hanya membahas proses penyisipan *file* di 3 bit terakhir.

5.1.3 Lingkungan Implementasi

Implementasi dilakukan pada lingkungan perangkat keras dan lingkungan perangkat lunak.

1. Perangkat Keras

Perangkat keras yang digunakan mempunyai spesifikasi sebagai berikut:

- a. *Processor* Intel Core Duo 1,83 GHz
- b. *Memory* 1 GB
- c. *Harddisk* berkapasitas 40 GB

2. Perangkat Lunak

Perangkat lunak dalam implementasi ini menggunakan:

- a. Sistem Operasi *Windows XP Professional*
- b. Bahasa pemrograman *Microsoft Visual Basic 6.0*

5.1.4 Tampilan aplikasi

Aplikasi ini dirancang khusus untuk membantu dalam mengamankan data *file* penting dengan teknik steganografi. Aplikasi keamanan data menggunakan teknik steganografi *file* pada gambar ini secara umum diperlihatkan melalui tampilan menu utama sistem seperti pada Gambar 5.1.

Aplikasi ini terdiri dari beberapa *Form* yang mempunyai aturan tertentu yang harus dijalankan secara berurutan.



Gambar 5.1 Tampilan Menu Utama Aplikasi Steganografi *File* Pada Gambar

Menu berikutnya merupakan menu pemilihan bit untuk melakukan proses penyisipan *file*. Seperti pada gambar 5.2 berikut ini:



Gambar 5.2 Tampilan Menu Pemilihan Bit Penyisipan

Menu berikutnya merupakan menu penyisipan *file* yang terdiri dari pemiliha *file* yang akan disisipkan, pemilihan gambar sebagai media pembawa, dan proses penyisipan . Seperti pada gambar 5.3.



Gambar 5.3 Tampilan Menu Penyisipan

Menu berikutnya merupakan menu pengambilan *file* yang terdiri dari pemiliha *file* yang akan disisipkan, pemilihan gambar sebagai media pembawa, dan proses penyisipan . Seperti pada gambar 5.4.



Gambar 5.4 Tampilan Menu Pengambilan File

5.2 Pengujian Sistem

Pengujian sistem ini dilakukan pada lingkungan perangkat lunak dan perangkat keras sesuai dengan lingkungan implementasi.

1. Pengujian Menggunakan *Black Box*
2. Pengujian Berdasarkan *Fidelity*
3. Pengujian Berdasarkan *Recovery*
4. Pengujian Berdasarkan Waktu
5. Pengujian Berdasarkan *Peak Signal to Noise Ratio* (PSNR)
6. Pengujian Berdasarkan *Robustness*

5.2.1 Pengujian Menggunakan *Black Box*

Pengujian dilakukan dengan melihat tampilan masing-masing form yang ada dalam setiap urutan proses steganografi maupun pada proses ekstraksi hasil.

5.2.2 Pengujian Modul

Pengujian modul ini merupakan hasil pengujian implementasi aplikasi secara detail mengenai item-item yang terdapat pada setiap tampilan proses menyisipkan *file* dalam gambar.

5.2.2.1 Pengujian Modul Penyisipan *File*

Prekondisi

1. Sudah ada sumber file gambar didalam perangkat penyimpanan yang akan dilakukan pengujian

Tabel 5.1 Tabel Butir Uji Pengujian Penyisipan File

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
Pengujian tahap 1 Proses penyisipan <i>file</i> ke dalam gambar	Tampilan layar menu utama, ada dua fasilitas option, penyisipan <i>file</i> dalam gambar atau pengambilan <i>file</i> dalam gambar	1.Tekan tombol “Pilih” untuk memilih bit gambar yang akan disisipkan <i>file</i> 2.Tekan tombol “Pilih” untuk memilih <i>file</i> yang akan disisipkan	File yang akan disisipkan ke gambar, Gambar sebagai media pembawa ,Kata kunci penyisipan	Gambar hasil steganografi yang sudah berhasil disisipkan <i>file</i> dan tidak ada instruksi error	Gambar hasil steganografi yang sudah berhasil disisipkan <i>file</i> dan tidak ada instruksi error	Gambar hasil steganografi yang sudah berhasil disisipkan <i>file</i> dan tidak ada instruksi error	Diterima

		3. Tekan tombol “Pilih” untuk memilih gambar sebagai media pembawa 4. Tekan tombol “Penyisipan Data” untuk melakukan proses penyisipan data sesuai dengan bit yang ditentukan 5. Masukkan kata kunci					
--	--	---	--	--	--	--	--

5.2.2.2 Pengujian Modul Pengambilan *File*

Prekondisi

1. Sudah ada gambar hasil steganografi didalam perangkat penyimpanan yang akan dilakukan pengujian

Tabel 5.2 Tabel Butir Uji Pengujian Pengambilan File

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
Pengujian tahap 2 Proses pengambilan <i>file</i> dari gambar	Tampilan layar menu utama, ada dua fasilitas option, penyisipan <i>file</i> dalam gambar atau pengambilan <i>file</i> dalam gambar	1. Tekan tombol “Pilih” untuk memilih gambar sebagai media pembawa 2. Tekan tombol “Penyisipan Data” untuk melakukan proses penyisipan data sesuai dengan bit yang ditentukan 3. Masukkan kata kunci	Gambar hasil steganografi, Kata kunci	<i>File</i> yang diambil dari gambar hasil steganografi dan tidak ada instruksi error	<i>File</i> yang diambil dari gambar hasil steganografi dan tidak ada instruksi error	<i>File</i> yang diambil dari gambar hasil steganografi dan tidak ada instruksi error	Diterima

5.2.3 Pengujian Berdasarkan *Fidelity*

Pengujian dilakukan untuk melihat mutu citra penampung apakah mengalami perubahan atau tidak. Pengujian fidelity dilakukan dengan melihat perubahan besar file gambar pada setiap bit penyisipan. Pengujian besar file dilakukan pada 3 file gambar.

5.2.3.1 Pengujian Besar File Gambar Hasil Steganografi

Pengujian besar 3 file gambar bitmap yang telah melalui proses steganografi di 3 bit terendah, adapun hasil pengujian dapat dilihat pada tabel berikut:

Tabel 5.3 Tabel Pengujian Besar File Gambar Hasil Steganografi

Bit Terendah Yang Disisipkan	Gambar Citra Bitmap	Ukuran File Yang Disisipkan	Ukuran Citra (piksel x piksel)	Ukuran Citra Awal (MB)	Ukuran Citra Akhir (MB)
8	Picture1.bmp	5,381 KB	3888 x 2592	69,048,930 bytes	69,048,930 bytes
7	Picture2.bmp	5,381 KB	3888 x 2592	69,048,930 bytes	69,048,930 bytes
6	Picture3.bmp	5,381 KB	3888 x 2592	69,048,930 bytes	69,048,930 bytes



Gambar 5.5 Picture1.bmp, Picture2.bmp, Picture3.bmp

Pada citra 24-bit yang berukuran 5876 x 3917 *pixel*, satu *pixel* berukuran 3 *byte* (atau 1 *byte* untuk setiap komponen R, G, B), sehingga kita bisa menyisipkan pesan sebanyak $23.016.292 \times 3 \text{ bit} = 69.048.876 \text{ bit}$ atau $69.048.876 / 8 = 8.631.109 \text{ byte}$. Maka untuk gambar berukuran 5876 x 3917 *pixel*, dapat menyisipkan file sebesar $8.631.109 / 1024 = 8.428 \text{ KB}$ atau 8,23 MB.

Dari tabel diatas dapat dilihat besar *file* tidak mengalami perubahan karena proses penyisipan biner *file* ke dalam biner gambar menggunakan metode penggantian 3 bit terakhir sehingga kapasitas gambar sebelum dan sesudah disteganografi tidak mengalami perubahan yang berarti. Hal tersebut merupakan

salah satu kelebihan dari metode *Least Significant Bit* (LSB). Untuk selanjutnya, penjelasan tampilan pengujian besar *file* dapat dilihat pada lampiran A.

5.2.4 Pengujian Berdasarkan *Recovery*

Data yang disembunyikan harus dapat diungkapkan kembali (*recovery*), karena tujuan steganografi adalah *data hiding*, maka sewaktu-waktu data rahasia di dalam citra penampung harus dapat diambil kembali untuk digunakan lebih lanjut.

Tabel 5.4 Tabel Pengujian Recovery

N o.	Masukkan (Gambar hasil steganografi)	Keluaran (File yang telah disisipkan)	Hasil
1.	Picture1.bmp	Irfan_Sistem_Berbasis_Pengetahuan.pdf (5,381 KB)	Berhasil
2.	Music1.bmp	Cinta Putih.mp3 (3,533 KB)	Berhasil
3.	Sunset1.bmp	Water lilies.jpg (81,8 KB)	Berhasil



(1)



(2)

Gambar 5.6 (1)Picture1.bmp, Gambar(2)Music1.bmp



(3)

Gambar 5.7 (3)Sunset1.bmp

Pengujian dilakukan dengan menjalankan aplikasi ekstraksi hasil, dapat diambil kesimpulan bahwa pengungkapan data kembali berhasil dan aplikasi memenuhi syarat *recovery*. Untuk selanjutnya, tampilan hasil ekstraksi dapat dilihat dilampiran B.

5.2.5 Pengujian Berdasarkan Waktu Penyisipan

Pengujian dilakukan pada beberapa gambar yang disisipkan berbagai macam jenis *file*. Pengujian berdasarkan waktu penyisipan ini terbagi atas 3 tabel sesuai dengan bit penyisipan yang telah dilakukan. Beberapa tabel pengujian berikut dibawah ini:

Tabel 5.5 Tabel Pengujian Waktu Penyisipan di Bit ke-8

No.	Nama Gambar Pembawa	Ukuran Pixel	Jenis File yang Disisipkan	Ukuran File yang Disisipkan	Waktu Penyisipan File	Waktu Pengambilan File
1.	Gambar1.bmp	3888 x 2592	BAB IV.doc	3,690 KB	5 menit 52 detik	3 jam 8 menit 13 detik
2.	Gambar 2.bmp	3888 x 2592	suhadi-report.doc	1,056 KB	2 menit 43 detik	33 menit 48 detik
3.	Gambar3.bmp	3888 x 2592	BAB III.doc	635 KB	1 menit 5 detik	5 menit 3 detik
4.	IMG_1.bmp	4080 x 2720	Kita Selamany a.mp3	4,057 KB	15 menit 35 detik	8 jam 8 menit 21 detik
5.	IMG_2.bmp	4080 x 2720	DiaryDepr esiku.mp3	2,717 KB	7 menit 39 detik	3 jam 49 menit 11 detik
6.	IMG_3.bmp	4080 x 2070	NAF.mp3	1,249 KB	2 menit	42 menit



(1)



(2)

Gambar 5.8 (1)Gambar1.bmp, (2)IMG_1.bmp

Pada Gambar1.bmp, Gambar2.bmp, dan Gambar3.bmp memiliki jumlah pixel yang sama yaitu 3888 x 2592. Kapasitas maksimum file yang dapat disisipkan pada ketiga gambar tersebut adalah 3.690 KB atau 3,6 MB. Masing-masing gambar disisipkan *file document* dengan kapasitas yang berbeda. Pada

Gambar1.bmp disisipkan *file document* dengan kapasitas maksimum yaitu 3,690 KB, estimasi waktu penyisipannya adalah 5 menit 52 detik dan estimasi waktu pengambilan file adalah 3 jam 8 menit 13 detik. Gambar2.bmp disisipkan *file document* dengan kapasitas setengah dari kapasitas maksimum atau kapasitas menengah yaitu 1,056 KB, estimasi waktu penyisipannya adalah 2 menit 43 detik dan estimasi waktu pengambilan file adalah 33 menit 48 detik. Gambar3.bmp disisipkan *file document* dengan kapasitas kecil yaitu 635 KB, estimasi waktu penyisipannya adalah 1 menit 5 detik dan estimasi waktu pengambilan file adalah 5 menit 3 detik.

Pada gambar IMG_1.bmp, IMG_2.bmp, dan IMG_3.bmp memiliki pixel yang berbeda dari 3 gambar sebelumnya ,yaitu 4080 x 2720. Kapasitas maksimum file yang dapat disisipkan pada ketiga gambar tersebut adalah 4.064 KB atau 3,96 MB. Pada gambar IMG_1.bmp disisipkan file mp3 dengan kapasitas maksimum, yaitu 4,057 KB, estimasi waktu penyisipannya adalah 36 menit 35 detik dan 6 jam 8 menit 21 detik. Gambar IMG_2.bmp disisipkan file mp3 dengan kapasitas menengah, yaitu 2,717 KB, estimasi waktu penyisipannya adalah 7 menit 39 detik dan waktu pengambilan file adalah 3 jam 49 menit 11 detik. Gambar IMG_3.bmp disisipkan file MP3 dengan kapasitas kecil, yaitu 1,249 KB, estimasi waktu penyisipannya adalah 2 menit dan waktu pengambilannya adalah 42 menit.

Tabel 5.6 Tabel Pengujian Waktu Penyisipan di Bit ke-7

No.	Nama Gambar Pembawa (Carrier Image)	Ukuran Pixel	Jenis File yang Disisipkan	Ukuran File yang Disisipkan	Waktu Penyisipan File	Waktu Pengambilan File
1.	Picture 1.bmp	5876 x 3917	Sistem_Berba sis_ Pengetahuan .pdf	5,381 KB	9 menit 12 detik	4 jam 34 menit 17 detik
2.	Picture 2.bmp	5876 x 3917	Steganograph y.pdf	2,331 KB	3 menit 29 detik	1 jam 45 menit 22 detik
3.	Picture3.bmp	5876 x 3917	L2F304217_ MTA.pdf	981 KB	1 menit 30 detik	20 menit 5 detik
4.	Pacujalur1.b mp	4655 x 3491	AdaBand- Masih.mp3	5,861 KB	8 menit 45 detik	4 jam 15 menit 3 detik
5.	Pacujalur2.b mp	4655 x 3491	Bruno Mars – Billionaire.mp 3	3,028 KB	7 menit 3 detik	3 jam 22 menit 15 detik
6.	Pacujalur3.b mp	4655 x 3491	Naff - TerendaP Laraku.mp3	1,249 KB	3 menit 54 detik	1 jam 33 menit 12 detik



(1)



(2)

Gambar 5.9 (1)Picture1.bmp, (2) Pacujalur1.bmp

Pada Picture 1.bmp, Picture 2.bmp, dan Picture3.bmp memiliki jumlah pixel yang sama yaitu 5876 x 3917. Kapasitas maksimum file yang dapat disisipkan pada ketiga gambar tersebut adalah 8.428 KB atau 8,23 MB. Masing-masing gambar disisipkan file .pdf dengan kapasitas yang berbeda. Pada Picture_1.bmp disisipkan file .pdf dengan kapasitas mendekati kapasitas

maksimum yaitu 5,381 KB, estimasi waktu penyisipannya adalah 9 menit 12 detik dan estimasi waktu pengambilan file adalah 4 jam 34 menit 17 detik. Picture_2.bmp disisipkan file .pdf dengan kapasitas menengah yaitu 2,331 KB, estimasi waktu penyisipannya adalah 3 menit 29 detik dan estimasi waktu pengambilan file adalah 1 jam 45 menit 22 detik. Picture_3.bmp disisipkan file .pdf dengan kapasitas kecil yaitu 981 KB, estimasi waktu penyisipannya adalah 1 menit 30 detik dan estimasi waktu pengambilan file adalah 20 menit 5 detik.

Pada gambar Pacujalur1.bmp, Pacujalur2.bmp, dan Pacujalur2.bmp memiliki pixel yang berbeda dari 3 gambar sebelumnya ,yaitu 4655 x 3491. Kapasitas maksimum file yang dapat disisipkan pada ketiga gambar tersebut adalah 5.951 KB atau 5,81 MB. Pada gambar Pacujalur1.bmp disisipkan file mp3 dengan kapasitas maksimum, yaitu 5,861 KB, estimasi waktu penyisipannya adalah 8 menit 45 detik dan estimasi waktu pengambilan file adalah 4 jam 15 menit 3 detik. Pada gambar Pacujalur2.bmp disisipkan file mp3 dengan kapasitas menengah, yaitu 3,028 KB, estimasi waktu penyisipannya adalah 7 menit 3 detik dan waktu pengambilan file adalah 3 jam 22 menit 15 detik. Pada gambar Pacujalur2.bmp disisipkan file MP3 dengan kapasitas kecil, yaitu 1,249 KB, estimasi waktu penyisipannya adalah 3 menit 54 detik dan waktu pengambilannya adalah 1 jam 33 menit 12 detik.

Tabel 5.7 Tabel Pengujian Waktu Penyisipan di Bit ke-6

No.	Nama Gambar Pembawa (Carrier Image)	Ukuran Pixel	Jenis File yang Disisipkan	Ukuran File yang Disisipkan	Waktu Penyisipan File	Waktu Pengambilan File
1.	IMG_9178.bmp	4752 x 3168	IMG_0682.JPEG	5,371KB	9 menit 54 detik	5 jam 25 menit 43 detik
2.	IMG_9177.bmp	4752 x 3168	IMG_0473.JPEG	3,070 KB	4menit 23 detik	2 jam 33 menit 48 detik
3.	IMG_9179.bmp	4752 x 3168	IMG_2227.JPEG	420 KB	1 menit 20 detik	38 menit 13 detik
4.	Wedding1.bmp	5000 x 3333	12072010.MP4	4,146 KB	11 menit 2 detik	6 jam 45 menit 21 detik
5.	Wedding2.bmp	5000 x 3333	08072010.MP4	3,253 KB	9 menit16 detik	4 jam 32 menit 12 detik
6.	Wedding3.bmp	5000 x 3333	12072010(001).MP4	1.332 KB	3 menit 59 detik	1 jam 44 menit 24 detik



(1)



(2)

Gambar 5.10 (1) IMG_9178.bmp, (2) Wedding1.bmp

Pada IMG_9178.bmp , IMG_9177.bmp, dan IMG_9179.bmp memiliki jumlah pixel yang sama yaitu 4752 x 3168. Kapasitas maksimum file yang dapat disisipkan pada ketiga gambar tersebut adalah 5513 KB atau 5,38 MB. Masing-masing gambar disisipkan file .jpeg dengan kapasitas yang berbeda. Pada gambar IMG_9178.bmp disisipkan file .jpeg dengan kapasitas mendekati kapasitas maksimum yaitu 5,371KB, estimasi waktu penyisipannya adalah 9 menit 54 detik

dan estimasi waktu pengambilan file adalah 5 jam 25 menit 43 detik. Gambar IMG_9177.bmp disisipkan file .jpeg dengan kapasitas menengah yaitu 3,070 KB, estimasi waktu penyisipannya adalah 4 menit 23 detik dan estimasi waktu pengambilan file adalah 2 jam 33 menit 48 detik. Gambar IMG_9179.bmp disisipkan file .jpeg dengan kapasitas kecil yaitu 420 KB, estimasi waktu penyisipannya adalah 1 menit 20 detik dan estimasi waktu pengambilan file adalah 38 menit 13 detik.

Pada gambar Wedding1.bmp, Wedding2.bmp, dan Wedding3.bmp memiliki pixel yang berbeda dari 3 gambar sebelumnya ,yaitu 5000 x 3333. Kapasitas maksimum file yang dapat disisipkan pada ketiga gambar tersebut adalah 6102 KB atau 5,95 MB. Pada gambar Wedding1.bmp disisipkan file MP4 dengan kapasitas maksimum, yaitu 4,146 KB, estimasi waktu penyisipannya adalah 11 menit 2 detik dan estimasi waktu pengambilan file adalah 6 jam 45 menit 21 detik. Pada gambar Wedding2.bmp disisipkan file MP4 dengan kapasitas menengah, yaitu 3,253 KB, estimasi waktu penyisipannya adalah 9 menit 16 detik dan waktu pengambilan file adalah 4 jam 32 menit 12 detik. Pada gambar Wedding3.bmp disisipkan file MP3 dengan kapasitas kecil, yaitu 1.332 KB, estimasi waktu penyisipannya adalah 3 menit 59 detik dan waktu pengambilannya adalah 1 jam 44 menit 24 detik.

Dari pengujian berdasarkan 3 tabel diatas, dapat disimpulkan bahwa apabila kapasitas file yang disisipkan mendekati atau sama dengan kapasitas maksimum, maka estimasi waktu yang dibutuhkan akan lebih lama. Dan sebaliknya, apabila kapasitas file yang disisipkan menengah atau lebih kecil dari kapasitas maksimum, maka estimasi waktu yang dibutuhkan lebih kecil juga.

Selain itu, juga dapat disimpulkan bahwa dalam proses pengambilan file membutuhkan waktu yang lebih lama daripada waktu penyisipan file. Hal ini dikarenakan proses yang terjadi didalam pengambilan file lebih rinci yaitu mulai dari pelacakan bit yang disisipkan kemudian mengambil bit-bit file yang disisipkan lalu menyatukan kembali bit-bit file menjadi sebuah data file yang sempurna.

5.2.6 Pengujian Berdasarkan *Peak Signal to Noise Ratio* (PSNR)

Pengujian dilakukan pada 3 gambar dimana gambar tersebut masing-masing telah disisipkan file dengan ukuran yang sama tetapi metode penyisipannya berbeda di 3 bit terendah yaitu bit 8, bit 7, dan bit 6. Hasil pengujian dapat dilihat di tabel berikut ini.

Tabel 5.8 Tabel Pengujian Citra dalam Nilai PSNR

No.	Bit yang Diganti	Ukuran Gambar Pembawa (<i>Carrier Image</i>)	Ukuran <i>File</i> yang Disisipkan	Nilai PSNR (db)	Nilai MSE
1	Bit 8	Bit 8.bmp (3888 x 2592)	2,73 MB	81.8734db	4.2242e
2	Bit 7	Bit 7.bmp (3888 x 2592)	2,73 MB	81.7536db	4.3423e
3	Bit 6	Bit 6.bmp (3888 x 2592)	2,73 MB	81.4536db	4.6528e

Ketiga gambar pada tabel yang memiliki ukuran yang sama yaitu 3888 x 2592 disisipkan *file document* yang juga berukuran sama, yaitu 2,73 MB. Hasil perhitungan nilai PSNR dari metode penyisipan bit ke-8, bit ke-7, dan bit ke-6 memiliki perbedaan yang tidak terlalu jauh. Nilai PSNR pada penyisipan bit ke-8 yaitu 54,9969 db, nilai PSNR pada penyisipan bit ke-7 yaitu 54.9909 db, dan nilai PSNR pada penyisipan bit ke-6 yaitu 48.2283 db. Semakin tinggi nilai PSNR berarti semakin bagus karena rasio *Signal-to-Noise* akan semakin tinggi. Pada umumnya, pemrosesan gambar dapat diterima oleh mata manusia jika nilai PSNR lebih besar dari 30 dB. Semakin besar PSNR, maka semakin baik kualitas gambar yang dihasilkan (Alatas, 2009). Maka dari perhitungan PSNR pada tabel diatas, gambar Bit 8.bmp, gambar Bit 7.bmp, dan gambar Bit 6.bmp mengalami penurunan kualitas gambar dikarenakan penyisipan bit yang berbeda disetiap gambar.

5.2.7 Pengujian Berdasarkan *Robustness*

Pengujian dilakukan dengan melihat data yang disembunyikan harus tahan terhadap manipulasi yang dilakukan pada citra penampung. Bila pada citra dilakukan operasi pengolahan citra, maka data yang disembunyikan tidak rusak. Hasil pengujian dapat dilihat pada tabel berikut ini:

Tabel 5.9 Tabel Pengujian Keberadaan File Dalam Gambar

No	File	Proses	Hasil
1.	Music3.bmp	Resize	Ekstraksi Gagal / Gambar Rusak
2.	IMG_9179.bmp	Crop	Ekstraksi Gagal / Gambar Rusak
3.	Gambar2.bmp	Rotasi 90 derajat	Ekstraksi Gagal / Gambar Rusak
4	Pacujalur2.bmp	Rotasi 180 derajat	Ekstraksi Gagal / Gambar Rusak

Dari hasil pengujian diketahui bahwa proses pengolahan citra dapat merusak karakter file yang berada dalam gambar bitmap, karena terjadinya perubahan letak biner. Hal ini terbukti dari proses ekstraksi hasil yang gagal dilakukan karena gambar telah rusak. Pengujian membuktikan salah satu kelemahan penggunaan metode LSB. Tampilan proses ekstraksi file yang gagal dapat dilihat dilampiran E.

5.3 Kesimpulan Pengujian

Setelah membandingkan antara hasil perancangan dan hasil yang didapat, maka dapat dilihat bahwa steganografi file ke citra bitmap menggunakan pengembangan metode *least significant bit modification (LSB)* di 3 bit terendah dapat dilakukan dengan sempurna. Tampilan aplikasi yang dihasilkan bersifat *user friendly* ketika diuji coba kepada beberapa *user*. Adapun yang dapat disimpulkan dari beberapa pengujian sebagai berikut:

1. Besar file gambar hasil steganografi tidak mengalami perubahan, karena proses penyisipan biner file ke dalam biner gambar menggunakan metode

penggantian bit terakhir sehingga kapasitas file gambar sebelum dan sesudah disteganografi tidak mengalami perubahan yang berarti.

2. Pengujian dilakukan dengan menjalankan aplikasi ekstraksi hasil, dapat diambil kesimpulan bahwa pengungkapan data kembali berhasil dan aplikasi memenuhi syarat *recovery*.
3. Pengujian dilakukan berdasarkan waktu, kapasitas file yang disisipkan mendekati atau sama dengan kapasitas maksimum, maka estimasi waktu yang dibutuhkan akan lebih lama. Dan sebaliknya, apabila kapasitas file yang disisipkan menengah atau lebih kecil dari kapasitas maksimum, maka estimasi waktu yang dibutuhkan lebih kecil juga. Selain itu, waktu pengambilan file lebih lama dibandingkan waktu penyisipan.
4. Pengujian dilakukan berdasarkan perhitungan PSNR, gambar yang telah disisipkan file pada bit ke-8, bit ke-7 dan bit ke-6 mengalami penurunan kualitas gambar.
5. Pengujian berdasarkan *robustness* dengan beberapa proses yaitu *resize*, *crop*, dan rotasi menimbulkan kegagalan dalam pengambilan file dikarenakan gambar telah rusak.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan analisa, perancangan, implementasi, dan pengujian pada aplikasi keamanan data menggunakan teknik steganografi file pada gambar dengan metode *Least Significant Bit (LSB)*, dapat diambil kesimpulan sebagai berikut:

1. Aplikasi steganografi pada citra digital dengan menerapkan pengembangan metode *Least Significant Bit* di 3 bit terakhir berhasil di implementasikan .
2. Besar file gambar sebelum dan sesudah proses steganografi tidak mengalami perubahan yang berarti.
3. Pengungkapan data kembali berhasil dilakukan dan aplikasi memenuhi syarat *recovery*.
4. Estimasi waktu yang dibutuhkan dalam proses steganografi berbeda-beda sesuai besar file yang disisipkan.
5. Perhitungan PSNR yang dilakukan pada gambar hasil steganografi membuktikan bahwa gambar mengalami kualitas penurunan sesuai posisi bit penyisipan.
6. Hasil steganografi tidak tahan terhadap pengolahan citra seperti *crop*, *resize*, dan rotasi.

6.2 Saran

Beberapa hal yang disarankan dalam pengembangan aplikasi steganografi pada citra digital dengan menerapkan metode *LSB* ini adalah sebagai berikut:

1. Pada aplikasi ini menggunakan gambar dengan format.bmp sebagai *file* penampung dan tidak menutup kemungkinan dikembangkan menggunakan gambar dengan format lain.
2. Aplikasi ini akan lebih baik jika dilengkapi dengan enkripsi dan deskripsi data file sebelum disisipkan ke dalam citra bitmap.
3. Dalam penyisipan data, aplikasi ini menggunakan metode *Least Significant Bit* (LSB), dan disarankan untuk mengembangkannya dengan metode lain untuk menghasilkan data steganografi yang lebih aman.

DAFTAR PUSTAKA

- Alatas, Putri. *Implementation Technique With Steganography LSB Method in Digital Images*. Universitas Gunadarma : Jakarta.2009
- Amira, Hapsari. *Studi Steganografi pada Image File*. Institut Teknologi Bandung: Bandung.2003
- Ariyus. D, *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi*, Andi Yogyakarta, Yogyakarta. 2008
- Munir, Rinaldi. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Informatika Bandung. 2004
- _____. *Kriptografi*, Informatika. Bandung.2006
- Pangaribuan, Ferry. *Pembangunan Aplikasi Penyembunyian Pesan yang Terenkripsi dengan Metode MARS pada Citra dengan Metode Zhang LSB Image*. Institut Teknologi Bandung:Bandung.2005
- Setiawan, Rachmansyah Budi. *Penggunaan Kriptografi dan Steganografi Berdasarkan Kebutuhan dan Karakteristik Keduanya*. Institut Teknologi Bandung:Bandung.2005

LAMPIRAN A

RINCIAN IMPLEMENTASI SISTEM

A.1 Pengujian Tampilan Menu Utama

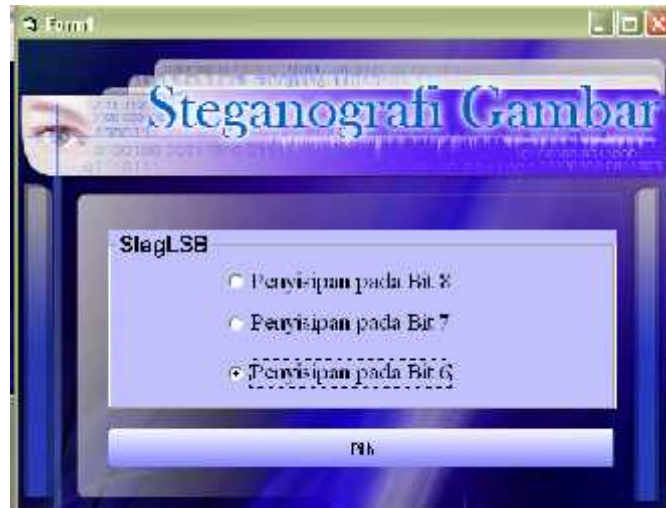
Pada tampilan awal diperlihatkan *menu* pilihan yang berfungsi untuk menentukan proses yang dilakukan yaitu proses penyisipan dan proses pengambilan data yang ditunjukkan oleh gambar A.1. Untuk melakukan proses penyisipan pilih *command button* ”Penyisipan File Dalam Gambar”.



Gambar A.1 Tampilan Pengujian Menu Utama

A.2 Pengujian Tampilan Menu Pemilihan Bit

Form ini merupakan tampilan untuk pemilihan bit gambar yang akan disisipkan *file*. Untuk menjalankan aplikasi, pilih *radio button*, lalu tekan tombol ”Pilih”, tampilan dapat dilihat di gambar A.2 berikut:



Gambar A.2 Tampilan Pengujian Menu Pemilihan Bit

A.3 Pengujian Tampilan Menu Penyisipan File

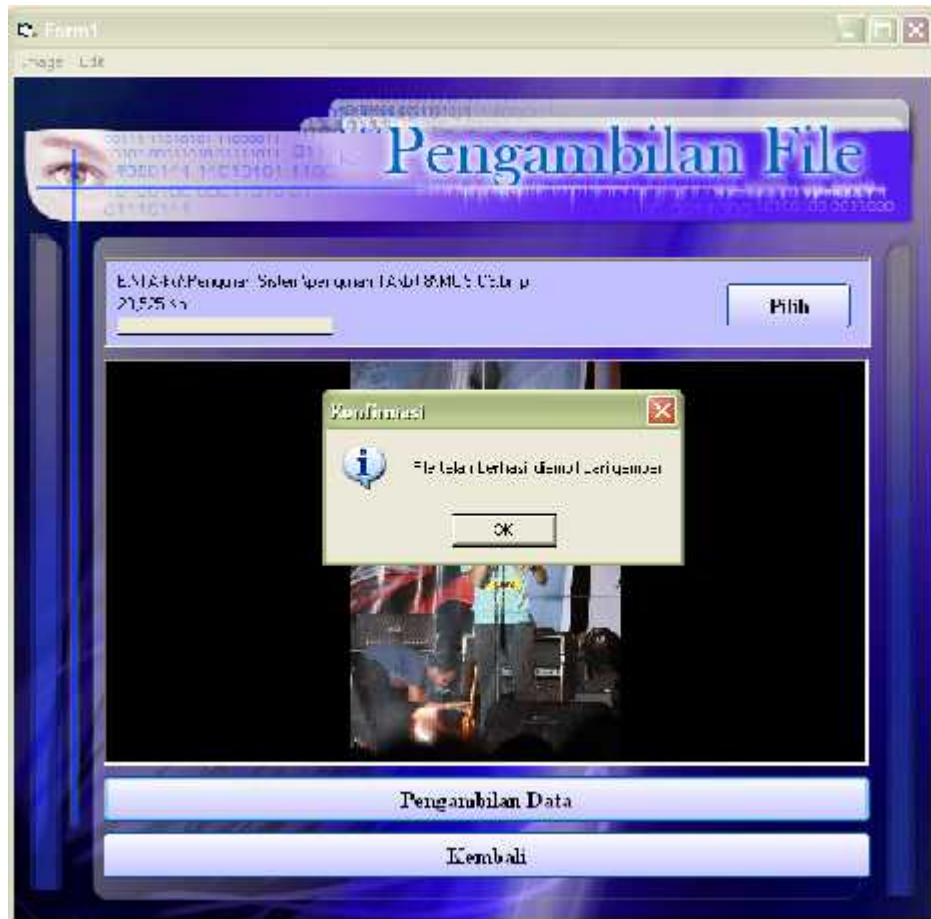
Pada *form* menu penyisipan *file* ini, *user* dapat memilih *file* yang akan disisipkan dan *user* dapat memilih gambar *bitmap* yang digunakan sebagai media pembawa pesan dengan menekan tombol “Pilih”. Kemudian *user* menekan tombol “Penyisipan Data” untuk melakukan proses penyisipan. Tampilan dapat dilihat pada gambar A.3 berikut ini :



Gambar A.3 Tampilan Pengujian Menu Penyisipan Data

A.4 Pengujian Tampilan Proses Ekstraksi Hasil.

Pada tampilan awal diperlihatkan *menu* pilihan yang berfungsi untuk menentukan proses yang dilakukan baik proses penyisipan maupun ekstraksi yang ditunjukkan oleh gambar A.4 untuk melakukan proses ekstraksi pilih *command button* "Pengambilan File dari Gambar".



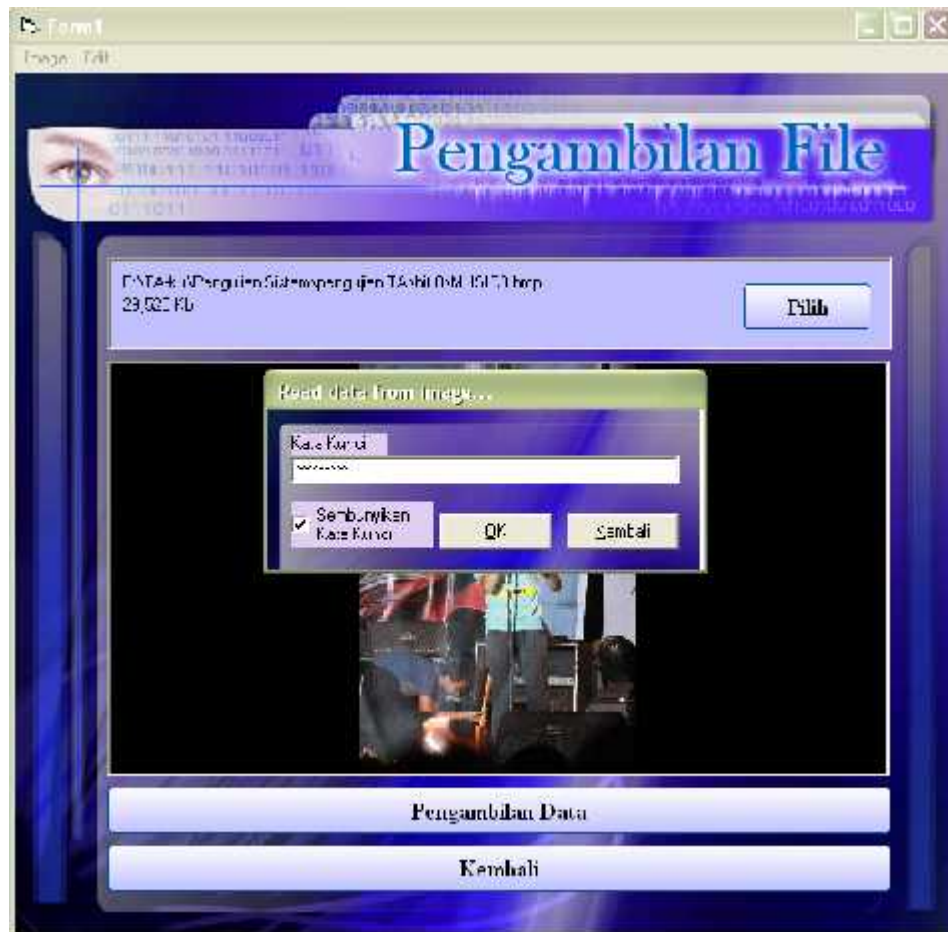
Gambar A.4 Tampilan Pengujian Menu Pengambilan Data

A.5 Pengujian Tampilan Input Kata Kunci

Pada *Form* ini *user* harus mengisi kolom kata kunci dengan memasukkan kata kunci yang diperoleh setelah melakukan proses penyisipan *file* kedalam gambar. Jika *user* salah memasukkan kata kunci yang diminta, maka *file* rahasia yang terdapat dalam gambar hasil steganografi tidak akan didapatkan. Tampilan aplikasi input kata kunci dapat dilihat pada gambar A.5 berikut.



Gambar A.5 Tampilan Pengujian Input Kata Kunci Penyisipan Data



Gambar A.6 Tampilan Pengujian Input Kata Kunci Pengambilan Data

LAMPIRAN B

PENGUJIAN SISTEM BERDASARKAN *FIDELITY*

Pengujian dilakukan untuk melihat mutu citra penampung apakah mengalami perubahan atau tidak. Pengujian fidelity dilakukan dengan melihat perubahan besar file gambar pada setiap bit penyisipan. Pengujian besar file dilakukan pada 3 file gambar.

B.1 Pengujian Besar File Gambar Sebelum Steganografi



Gambar B.1 Tampilan Propertis Gambar Sebelum Steganografi

B.2 Pengujian Besar File Gambar Sesudah Steganografi



Gambar B.2 Tampilan Propertis Gambar Sesudah Steganografi

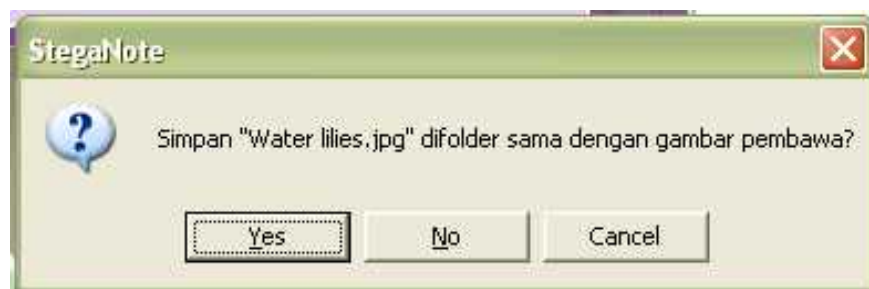
LAMPIRAN C

PENGUJIAN SISTEM BERDASARKAN *RECOVERY*

Data yang disembunyikan harus dapat diungkapkan kembali (*recovery*), karena tujuan steganografi adalah *data hiding*, maka sewaktu-waktu data rahasia di dalam citra penampung harus dapat diambil kembali untuk digunakan lebih lanjut. Pengujian dilakukan dengan menjalankan aplikasi ekstraksi hasil, dapat diambil kesimpulan bahwa pengungkapan data kembali berhasil dan aplikasi memenuhi syarat *recovery*.



Gambar C.1 Tampilan Pesan Ekstraksi



Gambar C.2 Tampilan Hasil Ekstraksi



Gambar C.3 Tampilan Hasil Ekstraksi

LAMPIRAN D

PENGUJIAN SISTEM BERDASARKAN PERHITUNGAN PSNR

Pada tabel dibawah ini merupakan perhitungan nilai PSNR dan MSE pada gambar-gambar yang telah melalui proses steganografi. Hasil pengujian dapat dilihat di tabel berikut ini:

No.	Ukuran Gambar Pembawa (<i>Carrier Image</i>)	Ukuran <i>File</i> yang Disisipkan	Nilai PSNR (db)	Nilai MSE
1	Gambar1. bmp (3888 x 2592)	3,690 KB	52.5702db	0.3598e
2	Gambar 2.bmp (3888 x 2592)	1,056 KB	57.4927db	0.1158e
3	Gambar3. bmp (3888 x 2592)	635 KB	60.9966db	0.0517e
4	IMG_1.bmp (4080 x 2720)	4,057 KB	52.0490db	0.4057e
5	IMG_2.bmp (4080 x 2720)	2,717 KB	52.5207db	0.3639e
6	IMG_3.bmp (4080 x 2070)	1,249 KB	53.0010db	0.256e
7	Picture 1.bmp (5876 x 3917)	5,381 KB	51.4778db	0.4627e
8	Picture 2.bmp (5876 x 3917)	2,331 KB	52.6957db	0.3495e
9	Picture3.bmp (5876 x 3917)	981 KB	53.3645db	0.2997e
10	Pacujalur1.bmp (4655 x 3491)	5,861 KB	51.9624db	0.4138e
11	Pacujalur1.bmp (4655 x 3491)	3,028 KB	52.768db	0.3231e
12	Pacujalur1.bmp (4655 x 3491)	1,249 KB	53.487db	0.2887e
13	IMG_9178.bmp (4752 x 3168)	5,371KB	52.9613db	0.3288e
14	IMG_9177.bmp (4752 x 3168)	3,070 KB	52.1235db	0.3988e
15	IMG_9179.bmp (4752 x 3168)	420 KB	63.1712db	0.0313e
16	Wedding1.bmp (5000 x 3333)	4,146 KB	53.6414db	0.2811e
17	Wedding2.bmp (5000 x 3333)	3,253 KB	54.7188db	0.2194e

18	Wedding3.bmp (5000 x 3333)	1.332 KB	58.5776db	0.0902e
19	Picture1.bmp (5876 x 3917)	5,381 KB	51.4778db	0.4627e
20	Music1.bmp (2592 x 3888)	3,533 KB	52.5202db	0.3594e
21	Sunset1.bmp (800 x 600)	81,8 KB	69.0616db	0.0081e